This is the document that completely defines the specifications of a proposed system. This is the basic document, which will be used as the basis for implementation.

The paragraphs written in the “Comment” style are for the benefit of the person writing the document and should be removed before the document is finalized.

In order to gain technical and methodological background refer to the following books:

- Applying Use Cases by Geri Schneider, Jason P. Winters
- Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design by Craig Larman
- Object Oriented Analysis and Design with Applications by Grady Booch

**Version:** <>

**Date:** <>

Prepared by:
**REVISION CHART**

This chart contains a history of this document’s revisions. The entries below are provided solely for illustration purposes. Those entries should be deleted until the revision/s they refer to have actually been created.

The document itself should be stored in revision control, and a brief description of each version should be entered in the Revision Control System. A brief description can be repeated in this section. Revisions need not be described elsewhere in the document, unless they explain the document.

<table>
<thead>
<tr>
<th>Version</th>
<th>Primary Author(s)</th>
<th>Description of Version</th>
<th>Date Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft</td>
<td>TBD</td>
<td>Initial draft created for distribution and review comments</td>
<td>(To be decided)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TBD</td>
</tr>
<tr>
<td>Preliminary</td>
<td>TBD</td>
<td>Second draft incorporating initial review comments, distributed for final review</td>
<td>TBD</td>
</tr>
<tr>
<td>Final</td>
<td>TBD</td>
<td>First complete draft, which is placed under change control</td>
<td>TBD</td>
</tr>
<tr>
<td>Revision 1</td>
<td>TBD</td>
<td>Revised draft, revised according to the change control process and maintained under change control</td>
<td>TBD</td>
</tr>
<tr>
<td>Revision 2</td>
<td>TBD</td>
<td>Revised draft, revised according to the change control process and maintained under change control</td>
<td>TBD</td>
</tr>
<tr>
<td>Etc.</td>
<td>TBD</td>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>
The preface contains an introduction to the document. It is optional and can be deleted if desired.
New paragraphs formatted as Heading 1, Heading 2, and Heading 3 will be added to the table automatically. To update this table of contents in Microsoft Word, put the cursor anywhere in the table and press F9. If you want the table to be easy to maintain, do not change it manually.

1. INTRODUCTION ..............................................................................................................................................6
  1.1 PROJECT OVERVIEW......................................................................................................................................6
  1.2 PROBLEM STATEMENT......................................................................................................................................6
  1.3 CUSTOMER........................................................................................................................................................6
  1.4 AFFECTED GROUPS........................................................................................................................................6
  1.5 ASSUMPTIONS.....................................................................................................................................................6
  1.6 DEPENDENCIES/ EXTERNAL SYSTEMS........................................................................................................7
  1.7 DEFINITIONS AND ACRONYMS.......................................................................................................................7
  1.8 REFERENCE/ SOURCE DOCUMENTS................................................................................................................7
  1.9 GOALS................................................................................................................................................................7

2. SYSTEM FUNCTIONS/ FUNCTIONAL REQUIREMENTS .................................................................................8

3. SYSTEM ARCHITECTURE .............................................................................................................................10
  3.1 SYSTEM ARCHITECTURE DIAGRAM............................................................................................................10

4. USE CASE MODEL .........................................................................................................................................11
  4.1 LIST OF ACTORS.............................................................................................................................................11
  4.2 LIST OF USE CASES......................................................................................................................................11
  4.3 USE CASE DIAGRAM...................................................................................................................................12
  4.4 USE CASE BUY ITEM..................................................................................................................................13
  4.5 SYSTEM SEQUENCE DIAGRAMS...................................................................................................................14
  4.6 USER INTERFACE.........................................................................................................................................15
  4.7 DATA DICTIONARY.........................................................................................................................................15

5. CONCEPTUAL MODEL ......................................................................................................................................16

6. GLOSSARY ........................................................................................................................................................17

7. INDEX ...............................................................................................................................................................18

8. APPENDICES......................................................................................................................................................19
LIST OF FIGURES

New figures that are given captions using the Caption paragraph style will be added to the table automatically. To update this table of contents in Microsoft Word, put the cursor anywhere in the table and press F9. If you want the table to be easy to maintain, do not change it manually.

This section can be deleted if the document contains no figures or if otherwise desired.

Figure 1 System Architecture ................................................................. 10
Figure 2 System Level Use Case Diagram .................................................. 12
Figure 3 Conceptual Model........................................................................ 16
1. **INTRODUCTION**

*This section should describe the project and the software product being to be built. No text is necessary between the heading above and the heading below unless otherwise desired.*

1.1 **Project Overview**

*Give a short summary of the project objective and the system to be analyzed*

Functional specifications are a description of needs or desires for a product. Identify and document what is really needed, in a form that clearly communicates to the client and to development team members. Define the requirements unambiguously, so that the risks are identified and there are no surprises when the product is finally delivered.

Following are the sample artifacts for this section:

- Problems or Overview Statement
- Customer
- Goals
- System functions
- System attributes

1.2 **Problem Statement**

*The purpose of this project is to …*

*The problem statement should be brief, comprising of no more than 50 words*

1.3 **Customer**

*A brief description of the client. The organization, its products/services etc.*

*e.g.*

*Company X offers solutions for companies who want to establish a portal on the Trading Net and those who want to host portals for others. Company X was founded in 1992 and went public in 1998.*

1.4 **Affected Groups**

*Those impacted by the deployment of the system. This can be a simple list as well as a bulleted one with short explanations.*

*e.g.*

- Sales staff
- Cashiers

*This may include users as well as support groups*

1.5 **Assumptions**

*Things we assume will be true.*
e.g.:

- We will receive all necessary technical support from the engineers at cMeRun, Select and Mellon Bank to help design the interfaces between their systems and enGyro.
- All database maintenance will be handled by the client.
- There will be no real-time interfacing with any accounting systems.

1.6 Dependencies/ External Systems

*Systems and/or products, this project depends upon for its completion.

* e.g.

- Cyber Cash

1.7 Definitions and Acronyms

*Provide definitions or references to all the definitions of the special terms and acronyms used within this document.*

1.8 Reference/ Source Documents

*Provide references to all documents that have been consulted during the analysis phase.*

1.9 Goals

*This brief section should focus on what the client wants to achieve. It must enumerate the objectives of the top management and what it hopes to accomplish from the proposed system.*
2. **SYSTEM FUNCTIONS/ FUNCTIONAL REQUIREMENTS**

This section is can be skipped, if Requirement Specifications document has been developed for the project. Otherwise this section is mandatory.

This section may contain

- end user, operator, support, or integration functions,
- performance requirements,
- design constraints,
- programming language, and
- interface requirements.

System functions are descriptions of what a system is supposed to do. They should be identified and listed in logical cohesive groups, with their category (priority) assigned. These system functions will be identified as a result of the requirement gathering process conducted with the client. However, in some cases, prior to the development of the Functional Specifications the requirements may already have been listed in a document: if this is so then a reference to the document may suffice.

To verify that some \( X \) is indeed a system function; it should make sense in the following sentence:

The system should do < \( X \)>

The table below gives an example of how system functions can be listed:

- The Functions column gives a brief one-line description of the required functionality.
- The Category column refers to the status of the functionality for the proposed system. The options for the Category are defined below.
- The Attribute column defines the system characteristics. The Details and Constraints column specifies the conditions within which the attribute is applicable. Section 1.12 defines the default Attributes and the related Constraints. In case, the default conditions are to be over-ridden then the conditions can be defined in this table.

Function Categories

<table>
<thead>
<tr>
<th>Function Category</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evident</td>
<td>Should perform, and user should be cognizant that it is performed.</td>
</tr>
<tr>
<td>Hidden</td>
<td>Should perform, but not be visible to users. This is true of many underlying technical services, such as save information in a persistent storage mechanism. Hidden functions are often missed during the requirements gathering process.</td>
</tr>
<tr>
<td>Frill</td>
<td>Optional; adding it does not significantly affect cost or other functions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ref#</th>
<th>Functions</th>
<th>Category</th>
<th>Attribute</th>
<th>Details &amp; Boundary Constraints</th>
</tr>
</thead>
</table>
| R1.1 | Record the underway sale – the items purchased | Evident | System Response time | Price listing within 3 seconds  
Availabilty agreement in less than 10 sec |
| R1.2 | Reduce inventory quantities when a sale is committed | Hidden | Concurrent user load |                                      |
System Attributes/ Nonfunctional Requirements

System attributes are nonfunctional system qualities – such as ease of use. System attributes are characteristics of the system; they are not functions.

System attributes have a possible set of Attribute Details, which tend to be discrete, fuzzy, symbolic values of the attribute, such as:

- response time = psychologically appropriate
- interface metaphor = graphical, browser-based

Some system attributes may also have Attribute Boundary Constraints, which are mandatory boundary conditions, usually on a numeric range of values of an attribute, such as:

- response time = five seconds maximum

In this section the Category column indicates whether or not the attribute is critical for the operation of the system.

The Category can take two options:

- Optional
- Mandatory

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Details and Boundary Constraints</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response time</td>
<td>(Boundary constraint) When recording a sold item, the description and price will appear within 5 seconds</td>
<td>Optional</td>
</tr>
<tr>
<td>Concurrent User Load</td>
<td>A minimum of 10 users connected simultaneously</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>
3. **SYSTEM ARCHITECTURE**

Describe the system architecture, or simply provide the architecture diagram. For School system it may include web based front end, webservers, database etc. Don’t worry too much about it just give a simple diagram of a typical web based project.

3.1 **System Architecture Diagram**

![System Architecture Diagram](image_url)

Figure 1 System Architecture
4. USE CASE MODEL

Describe the following items:

- Actors & use cases
- Use case diagrams
- High level, essential use cases

No text is necessary between the heading above and the heading below unless otherwise desired.

4.1 List of Actors

Define the system boundary and list all actors with the use cases.

For example:

- **Cashier**: this person performs all the financial activities
- **Account Manager**: this person supervises all financial activities

4.2 List of Use Cases

List all the use cases, with a brief description (should not exceed two lines):

- **Buy Item**: captures a sale and its payment
- **Log In**: allow user to provide account information and access the restricted services
4.3 Use Case Diagram

Create the system level use case diagram

Figure 2 System Level Use Case Diagram
4.4 Use Case Buy Item

Document each use case. This can completed using the tables provided below:

**Section: Main**

Name: Buy Item  
Actors: Customer, Cashier  
Purpose: Capture a sale and its payment.  
Description: A customer arrives at a checkout with items to purchase. The cashier records the purchase items and collects a payment. On completion, the customer leaves with the items.

Cross References: Functions: R1.1, R1.2  
Use Cases: Cashier must have completed the Log In use case. This is a reference to the System Functions as described in Section 1.10

**Pre-Conditions**  
Assumption about the state of the system before execution of the operation

**Successful Post-Conditions**  
State of the system after completion of the operation.

**Failure Post-Conditions**  
State of the system after completion of the operation.

<table>
<thead>
<tr>
<th>Typical Course of Events</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor Action</strong></td>
<td><strong>System Response</strong></td>
</tr>
<tr>
<td>1 This use case begins when a customer arrives at the Point of Sale checkout with items to purchase.</td>
<td></td>
</tr>
<tr>
<td>2 The cashier records each item</td>
<td>3 Determines the item price and adds the item information to the running sales transaction.</td>
</tr>
<tr>
<td>4 ...</td>
<td>5 ...</td>
</tr>
</tbody>
</table>
| 7 Customer selects payment type:  
  a. If cash payment, see section Pay by Cash  
  b. If credit payment, see section Pay by Credit | 8 Logs the completed sale  
  9 Updates inventory levels |
| 11 Cashier gives the receipt to the customer | 10 Generates a receipt |
| 12 The customer leaves with the items purchased |  |

**Alternative Course**

*Step 2: Invalid item identifier entered. Indicate error.*
Step 7: 

Customer could not pay. Cancel sales transaction

Section: Pay by Cash

Typical Course of Events

<table>
<thead>
<tr>
<th>Actor Action</th>
<th>System Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 The customer makes a cash payment</td>
<td></td>
</tr>
<tr>
<td>2 The cashier records the cash tendered</td>
<td></td>
</tr>
<tr>
<td>3 Presents the balance due back to the customer, if any.</td>
<td></td>
</tr>
<tr>
<td>4 The Cashier deposits the cash received and extracts the balance owing and gives it to the customer</td>
<td></td>
</tr>
</tbody>
</table>

Alternative Courses

Step 1: Customer does not have sufficient cash, may cancel sale or initiate another payment method.

Step 4: Cash drawer does not contain sufficient cash to pay balance.

4.5 System Sequence Diagrams

This is an optional section. It may help when the Typical Course Of Events (Section 3.4) is too detailed to clarify the flow properly.

A system sequence diagram is a picture that shows, for a particular scenario of a use case, the events that external actors generate, their order, and intersystem events. All systems are treated as a black box; the emphasis of the diagram is events that cross the system boundary from actor to systems.

A system sequence diagram should be completed for the typical course of events of the use case, and possibly others, for the most interesting alternative courses.
4.6 User Interface
This section may be used to provide screenshots of the application to give an idea of how the GUI will appear.

4.7 Data Dictionary
This section may be used to provide the details of interface elements that are present on the screenshots.

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Type</th>
<th>Validation</th>
<th>Mandatory</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5. **CONCEPTUAL MODEL**

Draw your conceptual model after writing all your use cases. This is an optional section, which will be applicable when there are a large number of entities.

A conceptual model is a representation of concepts in a problem domain. It is illustrated with a set of static structure diagrams. It can represent the following:

- Concepts
- Associations between concepts
- Attributes of concepts

A new example consistent with the Accounting System example used in this document should be provided.

![Conceptual Model Diagram]

**Figure 3 Conceptual Model**

*concept reacts differently to events depending on their state, it is considered as state-dependent concept or type. Create state diagrams only for state-dependent concepts with complex behavior.*

*A relevant example needs to be inserted here.*
6. GLOSSARY

A glossary or model dictionary lists and defines all the terms that require clarification in order to improve communication and reduce the risk of misunderstanding.

Record domain or business terms, rules, concepts, etc. in the glossary

<table>
<thead>
<tr>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS</td>
</tr>
<tr>
<td>Div</td>
</tr>
<tr>
<td>Package</td>
</tr>
</tbody>
</table>
7. **INDEX**

The index is optional. If the document is made available in electronic form, readers can search for terms electronically.
8. APPENDICES

Include supporting detail that would be too distracting to include in the main body of the document.