<Company Name>

<Project Name>
Software Requirements Specification

Version <1.0>

[Note: Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Info>Properties>Advanced Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting all (e.g., Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

Date	**Version**	**Description**	**Author**
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

1. Introduction 5

1.1 The Purpose of the <*Cafeteria Ordering System (COS)*> 5
1.2 The Purpose of this Document 5
1.3 Document Conventions 5
1.4 Product Scope 6
1.5 Definitions, Acronyms, and Abbreviations 6
1.6 References 6

2. Overall Description 8

2.1 Product Perspective 8
2.2 User Classes and Characteristics 8
2.3 Operating Environment 9
2.4 Design and Implementation Constraints 9
2.5 Assumptions and Dependencies 10

3. Specific Requirements 11

3.1 Use Case 1 (Employee Orders Meals) 11
3.1.1 Description 11
3.1.2 Functional Requirements 11
3.2 Use Case 2 (Change Meal Order) 13
3.3 Use Case 3 (Cancel a Meal Order) 13
3.4 Use Case 4 (View Menu) 13
3.5 Use Case 5 (Register for Payroll Deduction) 13
3.6 Use Case 6 (Unregister for Payroll Deduction) 13

4. Data Requirements 14

4.1 Logical Data Model 14
4.2 Data Dictionary 14
4.3 Reports 15
4.4 Data Acquisition, Integrity, Retention, and Disposal 16

5. External Interface Requirements 17

5.1 User Interfaces 17
5.2 Software Interfaces 17
5.3 Hardware Interfaces 18
5.4 Communications Interfaces 18

6. Quality Attributes 19

6.1 Usability 19
6.2 Performance 19
6.3 Security 19

6.4 Safety 19
6.5 Availability 20
6.6 Robustness 20
6.7 [Others as relevant] 20

7. Internationalization and Localization Requirements 21

8. Other Requirements 21

9. Appendix A: 22

Software Requirements Specification

The SRS states as completely as necessary the system's behaviors under various conditions, as well as desired system qualities such as performance, security, and usability etc.

Numerous audiences rely on this SRS. When you write SRS, keep the following audience in mind.

- *Customers, the marketing department, and sales staff need to know what product they can expect to be delivered.*
- *Project managers base their estimates of schedule, effort, and resources on the requirements.*
- *Software development teams need to know what to build.*
- *Testers use it to develop requirements-based tests, test plans, and test procedures.*
- *Maintenance and support staff use it to understand what each part of the product is supposed to do.*
- *Documentation writers base user manuals and help screens on the SRS and the user interface design.*
- *Training personnel use the SRS and user documentation to develop educational materials.*
- *Legal staff ensures that the requirements comply with applicable laws and regulations.*
- *Subcontractors base their work on—and can be legally held to—the specified requirements.*

1. Introduction

[The introduction of the ***Software Requirements Specification (SRS)*** *provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the* ***SRS****.]*

1.1 The Purpose of the <*Cafeteria Ordering System (COS)*>

[A requirements specification should open with a description of what the system is for: who wants it, and why? Who will use it? What is the business motivation behind it? Even when they have been answered already in the vision and scope document, there is a need to reiterate here in case people haven't read vision and scope. The size of the system purpose description varies depending upon the complexity and nature of the system. A single paragraph may suffice, though it should be a meaty paragraph; a single sentence is undoubtedly insufficient. Don't be lazy!]

1.2 The Purpose of this Document

[Specify the purpose of this ***Software Requirements Specification****. The* ***SRS*** *fully describes the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]*

Example:

This purpose of this document is to describe the functional and nonfunctional requirements for software release 1.0 of the Cafeteria Ordering System (COS). Its role is to describe the problem to be solved, not the solution: what the system must do, not how. This document is intended to be used by the members of the project team who will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are committed for release 1.0.

1.3 Document Conventions

<Describe any standards or typographical conventions used, including the meaning of specific text styles, highlighting, or notations. If you are manually labeling unique requirement identifiers, you might specify the format here for anyone who needs to add one later. The example below shows one way to format each of the requirement , please note, this is not the same as the format I introduced later on)>

Example:

This document contains a mixture of formal requirements and supporting information. The latter is descriptive only and does not form part of any formal delivery commitment. It is important to be able to distinguish the formal requirements from other information and to be able to unambiguously refer to each individual requirement. This is achieved by using the following standard format for the requirements themselves:

Requirement ID	Priority	Requirement Definition
	<<n>>	<<A clear statement of what is required>>

The meaning of each of these columns is as follows:

- *Column 1: Consultant Tom Gilb suggests a text-based hierarchical tagging scheme for labeling individual requirements. Consider this requirement: "The system shall ask the user to confirm any request to print more than 10 copies." This requirement might be tagged Print.ConfirmCopies. This indicates that it is part of the print function and relates to the number of copies to print. Hierarchical textual tags are structured, meaningful, and unaffected by adding, deleting, or moving other requirements. This sample functional requirements in section 3 illustrates this labeling technique.*
- *Column 2: Priority: This states how important the requirement is.*
- *Column 3: Definition: This is the description, in EARS syntax, that formally defines the requirement.*

Each requirement is a single, measurable objective. A requirement should not lump two or more things together, although there are situations where it is more practical to list several things at once rather than cause an explosion in the number of requirements (for example, when listing the fields that must appear on a particular screen).

1.4 Product Scope

[Provide a short description of the software being specified and its purpose. Relate the software to user or corporate goals and to business objectives and strategies. If a separate vision and scope or similar document is available, refer to it rather than duplicating its contents here. An SRS that specifies an incremental release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision. You might provide a high-level summary of the major features the release contains or the significant functions that it performs.]

Example:

The COS will permit Process Impact Ltd. employees to order meals from the company cafeteria online to be delivered to specified campus locations. A detailed description is available in the Cafeteria Ordering System Vision and Scope Document [1], along with the features that are scheduled for full or partial implementation in this release.

1.5 Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the ***Software Requirements Specification****. This information may be provided by reference to the project's Glossary.]*

1.6 References

[List any documents or other resources to which this SRS refers. Include hyperlinks to them if they are in a persistent location. These might include user interface style guides, contracts, standards, system requirements specifications, interface specifications, or the SRS for a related product. Provide enough information so that the reader can access each reference, including its title, author, version number, date, and source, storage location, or URL.]

Example:

1. Cafeteria Ordering System Vision and Scope Document,

www.processimpact.com/projects/COS/COS Vision and Scope.docx

2. Process Impact Intranet Development Standard, Version 1.3,

www.processimpact.com/corporate/standards/PI Intranet Development Standard.pdf

3. Process Impact Internet Application User Interface Standard, Version 2.0, www.processimpact.com/corporate/standards/PI Internet UI Standard.pdf

4. Process Impact Internet Application Business Rules, Version 2.0, www.processimpact.com/corporate/br/PI Business Rules.pdf

2. Overall Description

*[This section of the **Software Requirements Specification** describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as product perspective, product functions, user characteristics, constraints, assumptions and dependencies, and requirements subsets.]*

2.1 Product Perspective

[Describe the product's context and origin. Is it the next member of a growing product line, the next version of a mature system, a replacement for an existing application, or an entirely new product? If this SRS defines a component of a larger system, state how this software relates to the overall system and identify major interfaces between the two. Consider including visual models such as a context diagram to show the product's relationship to other systems. It other words, show how the system fits into the world around it: a local map showing where our territory ends and that of the neighbors' starts]

Example:

The Cafeteria Ordering System is a new software system that replaces the current manual and telephone processes for ordering and picking up meals in the Process Impact cafeteria. The context diagram in Figure C-2 illustrates the external entities and system interfaces for release 1.0. Note that the COS is denoted as circle in the middle. The system is expected to evolve over several releases, ultimately connecting to the Internet ordering services for several local restaurants and to credit and debit card authorization services.

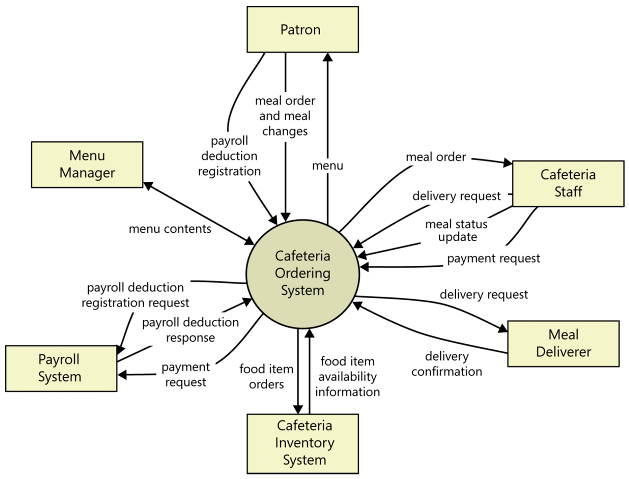

Patron
meal order and meal changes
payroll deduction registration
menu
Menu Manager
menu contents
meal order
Cafeteria Staff
delivery request
meal status update
payment request
Cafeteria Ordering System
payroll deduction registration request
payroll deduction response
delivery request
Meal Deliverer
Payroll System
payment request
food item orders
food item availability information
delivery confirmation
Cafeteria Inventory System

FIGURE C-2 Context diagram for release 1.0 of the Cafeteria Ordering System.

[The context diagram should be followed by a brief description of each component. After the diagram's explanatory notes, describe all the interfaces our system has with other system, it is OK to reference to external interface section in this document, be brief here.]

2.2 User Classes and Characteristics

[Identify the various user classes that you anticipate will use this product and describe their pertinent characteristics. Some requirements might pertain only to certain user classes. Identify the favored user classes. User classes represent a subset of the stakeholders described in the vision and scope document. User class descriptions are a reusable resource. If available, you can incorporate user class descriptions by simply pointing to them in a master user class catalog instead of duplicating information here.]

Example:

User class	*Description*

Patron (favored)	*A Patron is a Process Impact employee who wants to order meals to be delivered from the company cafeteria. There are about 600 potential Patrons, of which 300 are expected to use the COS an average of 5 times per week each. Patrons will sometimes order multiple meals for group events or guests. An estimated 60 percent of orders will be placed using the corporate intranet, with 40 percent of orders being placed from home or by smartphone or tablet apps.*
Cafeteria Staff	*The Process Impact cafeteria employs about 20 Cafeteria Staff who will receive orders from the COS, prepare meals, package them for delivery, and request delivery. Most of the Cafeteria Staff will need training in the use of the hardware and software for the COS.*
Menu Manager	*The Menu Manager is a cafeteria employee who establishes and maintains daily menus of the food items available from the cafeteria. Some menu items may not be available for delivery. The Menu Manager will also define the cafeteria's daily specials. The Menu Manager will need to edit existing menus periodically.*
Meal Deliverer	*As the Cafeteria Staff prepare orders for delivery, they will issue delivery requests to a Meal Deliverer's smartphone. The Meal Deliverer will pick up the food and deliver it to the Patron. A Meal Deliverer's other interactions with the COS will be to confirm that a meal was (or was not) delivered.*

2.3 Operating Environment

[Describe the environment in which the software will operate, including the hardware platform; operating systems and versions; geographical locations of users, servers, and databases; and organizations that host the related databases, servers, and websites. List any other software components or applications with which the system must peacefully coexist. If extensive technical infrastructure work needs to be performed in conjunction with developing the new system, consider creating a separate infrastructure requirements specification to detail that work.]

Example:

OE-1: The COS shall operate correctly with the following web browsers: Windows Internet Explorer versions 7, 8, and 9; Firefox versions 12 through 26; Google Chrome (all versions); and Apple Safari versions 4.0 through 8.0.

OE-2: The COS shall operate on a server running the current corporate-approved versions of Red Hat Linux and Apache HTTP Server.

OE-3: The COS shall permit user access from the corporate intranet; from a VPN Internet connection; and by Android and iOS smartphones and tablets.

2.4 Design and Implementation Constraints

[Describe any factors that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing or memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; programming language requirements or restrictions.]

Example:

CO-1: The system's design, code, and maintenance documentation shall conform to the Process Impact Intranet Development Standard, Version 1.3 [2].
CO-2: The system shall use the current corporate standard Oracle database engine.
CO-3: All HTML code shall conform to the HTML 5.0 standard.
CO-4: Oracle Java 11.0.5 must be used as the backend programming language.

2.5 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, reuse expectations, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors outside its control.>

Example:

AS-1: The cafeteria is open for breakfast, lunch, and supper every company business day in which employees are expected to be on site.

DE-1: The operation of the COS depends on changes being made in the Payroll System to accept payment requests for meals ordered with the COS.

DE-2: The operation of the COS depends on changes being made in the Cafeteria Inventory System to update the availability of food items as COS accepts meal orders.

3. Specific Requirements

*[This section of the **Software Requirements Specification** contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements and testers to test that the system satisfies those requirements. This template illustrates organizing the functional requirements for the product by use cases. You may prefer to organize this section by system features (the major services provided by the product), mode of operation, user class, object class, functional hierarchy, stimulus, response, or combinations of these, whatever makes the most logical sense for your product.]*

3.1 Use Case 1 (Employee Orders Meals)

3.1.1 Description

[Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority.]

Example:

A cafeteria Patron whose identity has been verified can order meals either to be delivered to a specified company location or to be picked up in the cafeteria. A Patron can cancel or change a meal order if it has not yet been prepared. Priority = High.

3.1.2 Functional Requirements

[Itemize the specific functional requirements associated with this use case/feature. These are the software capabilities that must be implemented for the user to carry out the feature's services or to perform a use case. Also describe how the product should respond to anticipated error conditions. Use "TBD" as a placeholder to indicate when necessary information is not yet available.]

Requirement ID	Priority	Requirement Definition
Order.Logon		When the Patron indicates she wants to order a meal, the COS shall confirm that the Patron is logged into the system. If the Patron is not logged into the system, the COS shall ask the Patron to log on or to exit.
Order.PayrollRegister		When the Patron is logged on, the COS shall confirm that the Patron is registered for meal payments by payroll deduction. If the Patron is not registered for payroll deduction, the COS shall give the Patron options to either register now and continue placing an order, or to continue anyway (in this case, the Patron cannot pick "delivery" and must pick up the order in the cafeteria and pay by cash) or to exit.
Order.Date.Prompt		Once the Patron is confirmed as registered, or unregistered but chooses to continue anyway, the COS shall display available meal dates and prompt the Patron for the meal date. (see BR-8) The COS shall allow the Patron to enter the meal date.
Order.Date.Cutoff		If the entered meal date is the current date (i.e., today) and the current time is after the order cutoff time, the COS shall inform the Patron that it is too late to place an order for today. The COS shall allow the Patron to either change the meal date or cancel the order.
Order.Deliver.Prompt		The COS shall prompt the Patron for delivery method: "delivery" or "pickup". If the Patron is unregistered, then "pickup" is the only option.
Order.Deliver.AvailableTimes		When "delivery" is indicated by the Patron and there are still

		delivery times for the meal date, the COS shall display the available delivery times for the meal date and ask the Patron to pick a time or change the order to be picked up in the cafeteria, or to cancel the order.
Order.Deliver.NoTimes		When "delivery" is indicated by the Patron but there are no available delivery times for the meal date, the COS shall notify the Patron. The COS shall allow the Patron to either change to pick up the order in the cafeteria or to change a different meal date or to cancel the order.
Order.Deliver.Location		When the Patron picks a delivery time, the COS shall ask the Patron to provide a valid delivery location.
Order.Menu.Display		After the Patron provides a delivery method and location, the COS shall display menu of available food items and the daily special for the entered meal date.
Order.Menu.Available		The COS shall display only those food items for which at least one unit is available in the cafeteria's inventory, and which can be delivered for the specified date.
Order.FoodSelection		The COS shall allow the Patron to select one or more food items from the menu. The COS shall allow the Patron to order one or more units of one food item. If the Patron orders more units of a food item than are presently in the cafeteria's inventory, the COS shall inform the Patron of the maximum number of units of that food item that he can order.
Order.MultipleMeals		The COS shall permit the Patron to order multiple identical meals, up to the fewest available units of any menu item in the order.
Order.Confirm.DisplayDetails		When the Patron indicates that she is done, the COS shall display the food items ordered, the individual food item prices, number of identical meals ordered, and the payment amount calculated per BR-12.
Order.Confirm		The COS shall allow the Patron to confirm the order.
Order.Modify		The COS shall allow the Patron to edit or cancel the order.
Order.Pay		When the Patron confirms the order, the COS shall ask the Patron to select a payment method.
Order.Pay.Delivery		If the meal is to be delivered, the Patron shall choose to pay by payroll deduction. (see BR-11)
Order.Pay.Pickup		If the meal is to be picked up in the cafeteria, the Patron shall choose to pay by payroll deduction or by cash at the time of pickup.
Order.Pay.Deduct		When the payroll deduction is selected, the COS shall issue a payment request to the external Payroll System.
Order.Pay.Deduct.OK		If the payment request is accepted, the COS shall display a message confirming acceptance of the order with a transaction number.
Order.Pay.Deduct.Reject		If the payment request is rejected, the COS shall display the reason for the rejection. The Patron shall either cancel the order, or change the payment method to cash and request to pick up the order at the cafeteria.
Order.Accepted		When the payment is accepted, the COS shall do the following as a single database transaction: ● Assign the next available meal order number to the meal and store the meal order with a status of "Accepted." ● Send a message to the Cafeteria Inventory System with

		the number of units of each food item in the order. ● Update the menu for the current order's order date to reflect any items that are now out of stock in the cafeteria inventory. ● Update the remaining available delivery times for the date of this order. ● Send an email message or text message (depending on the Patron's profile setting) to the Patron with the meal order and meal payment information. ● Send an email message to the Cafeteria Staff with the meal order information. ● If any step of Order.Done fails, the COS shall roll back the transaction and notify the user that the order was unsuccessful, along with the reason for failure.

3.2 Use Case 2 (Change Meal Order)

3.3 Use Case 3 (Cancel a Meal Order)

Functions requirements are like create, view, modify and delete meal subscriptions.

3.4 Use Case 4 (View Menu)

Functions requirements are like create, view, modify and delete cafeteria menus.

3.5 Use Case 5 (Register for Payroll Deduction)

View ingredient lists and nutritional information for cafeteria menu items.

3.6 Use Case 6 (Unregister for Payroll Deduction)

Provide system access through corporate intranet, smart phone, tablet, and outside Internet access by authorized employees

4. Data Requirements

[This section describes various aspects of the data that the system will consume as inputs, process in some fashion, or create as outputs.]

4.1 Logical Data Model

[A data model is a visual representation of the data objects and collections the system will process and the relationships between them. Include a data model for the business operations being addressed by the system, or a logical representation for the data that the system itself will manipulate. Data models are most commonly created as an entity-relationship (ER) diagram.]

Example:

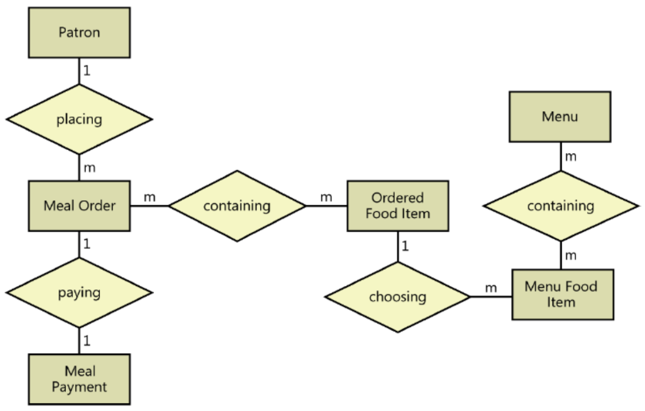
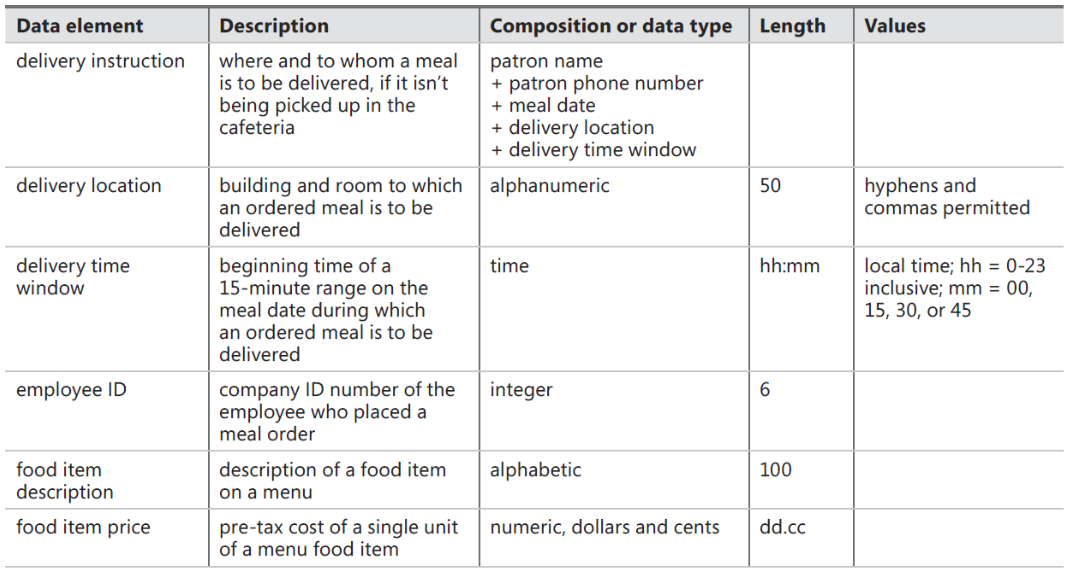


FIGURE C-3 Partial data model for release 1.0 of the Cafeteria Ordering System.

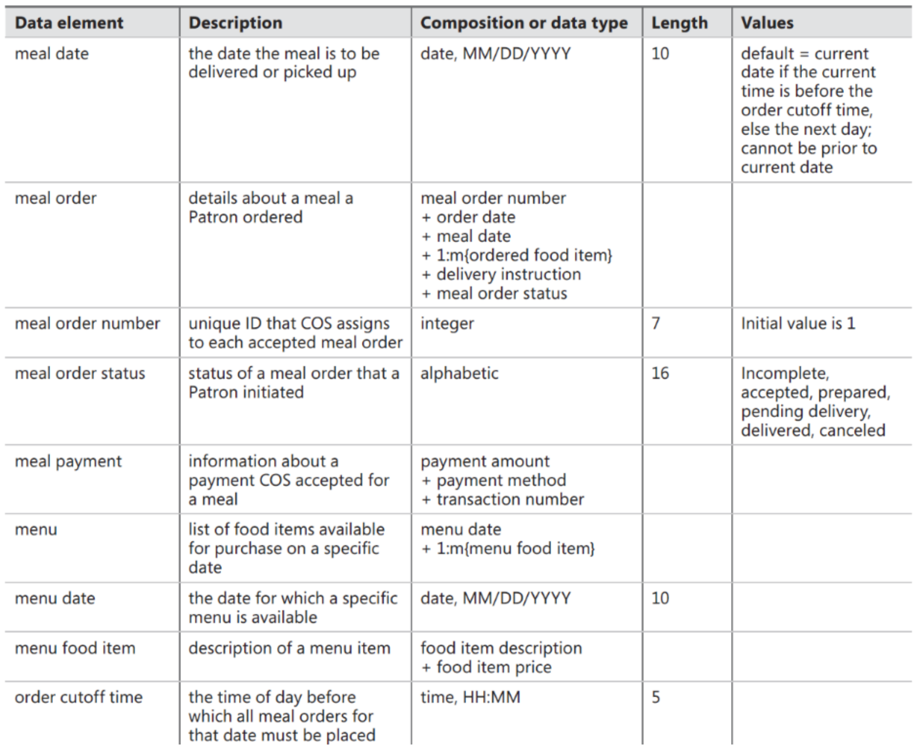
4.2 Data Dictionary

[The data dictionary defines the composition of data structures and the meaning, data type, length, format, and allowed values for the data elements that make up those structures. In many cases, you're better off storing the data dictionary as a separate artifact, rather than embedding it in the middle of an SRS. That also increases its reusability potential in other projects.]

Example:



Data element	Description	Composition or data type	Length	Values
delivery instruction	where and to whom a meal is to be delivered, if it isn't being picked up in the cafeteria	patron name + patron phone number + meal date + delivery location + delivery time window		
delivery location	building and room to which an ordered meal is to be delivered	alphanumeric	50	hyphens and commas permitted
delivery time window	beginning time of a 15-minute range on the meal date during which an ordered meal is to be delivered	time	hh:mm	local time; hh = 0-23 inclusive; mm = 00, 15, 30, or 45
employee ID	company ID number of the employee who placed a meal order	integer	6	
food item description	description of a food item on a menu	alphabetic	100	
food item price	pre-tax cost of a single unit of a menu food item	numeric, dollars and cents	dd.cc	

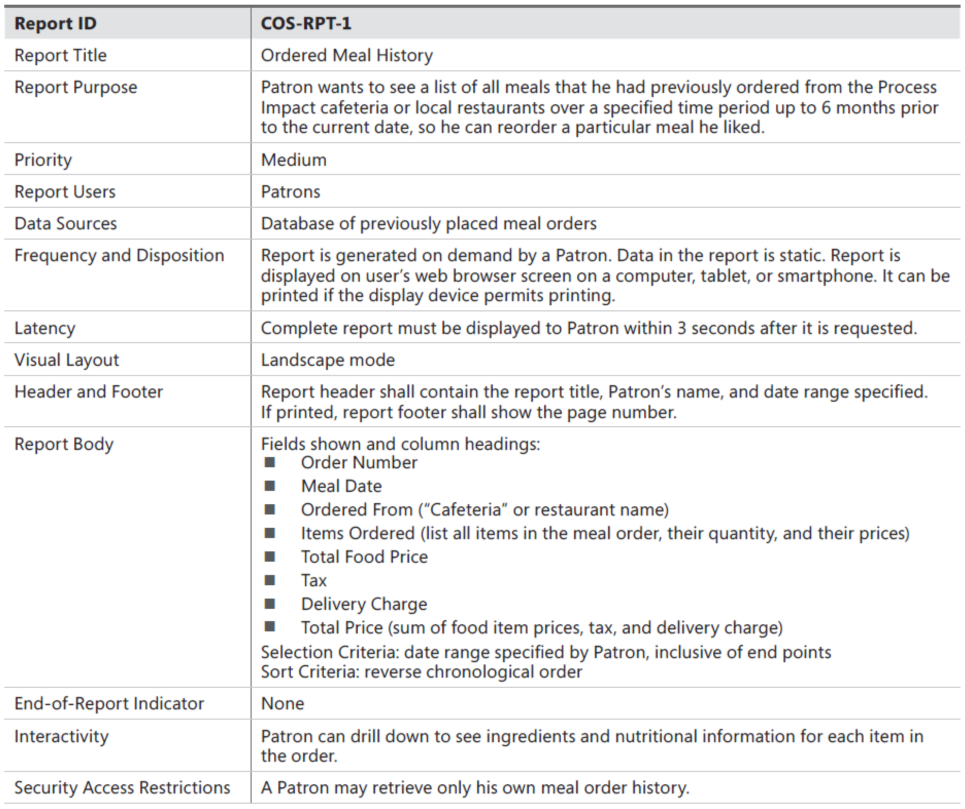


Data element	Description	Composition or data type	Length	Values
meal date	the date the meal is to be delivered or picked up	date, MM/DD/YYYY	10	default = current date if the current time is before the order cutoff time, else the next day; cannot be prior to current date
meal order	details about a meal a Patron ordered	meal order number + order date + meal date + 1:m{ordered food item} + delivery instruction + meal order status		
meal order number	unique ID that COS assigns to each accepted meal order	integer	7	Initial value is 1
meal order status	status of a meal order that a Patron initiated	alphabetic	16	Incomplete, accepted, prepared, pending delivery, delivered, canceled
meal payment	information about a payment COS accepted for a meal	payment amount + payment method + transaction number		
menu	list of food items available for purchase on a specific date	menu date + 1:m{menu food item}		
menu date	the date for which a specific menu is available	date, MM/DD/YYYY	10	
menu food item	description of a menu item	food item description + food item price		
order cutoff time	the time of day before which all meal orders for that date must be placed	time, HH:MM	5	

4.3 Reports

[If your application will generate any reports, identify them here and describe their characteristics. If a report must conform to a specific predefined layout you can specify that here as a constraint, perhaps with an example. Otherwise, focus on the logical descriptions of the report content, sort sequence, totaling levels, and so forth, deferring the detailed report layout to the design stage.]

Example:



Report ID	COS-RPT-1
Report Title	Ordered Meal History
Report Purpose	Patron wants to see a list of all meals that he had previously ordered from the Process Impact cafeteria or local restaurants over a specified time period up to 6 months prior to the current date, so he can reorder a particular meal he liked.
Priority	Medium
Report Users	Patrons
Data Sources	Database of previously placed meal orders
Frequency and Disposition	Report is generated on demand by a Patron. Data in the report is static. Report is displayed on user's web browser screen on a computer, tablet, or smartphone. It can be printed if the display device permits printing.
Latency	Complete report must be displayed to Patron within 3 seconds after it is requested.
Visual Layout	Landscape mode
Header and Footer	Report header shall contain the report title, Patron's name, and date range specified. If printed, report footer shall show the page number.
Report Body	Fields shown and column headings: ■ Order Number ■ Meal Date ■ Ordered From ("Cafeteria" or restaurant name) ■ Items Ordered (list all items in the meal order, their quantity, and their prices) ■ Total Food Price ■ Tax ■ Delivery Charge ■ Total Price (sum of food item prices, tax, and delivery charge) Selection Criteria: date range specified by Patron, inclusive of end points Sort Criteria: reverse chronological order
End-of-Report Indicator	None
Interactivity	Patron can drill down to see ingredients and nutritional information for each item in the order.
Security Access Restrictions	A Patron may retrieve only his own meal order history.

4.4 Data Acquisition, Integrity, Retention, and Disposal

[If relevant, describe how data is acquired and maintained. State any requirements regarding the need to protect the integrity of the system's data. Identify any specific techniques that are necessary, such as backups, checkpointing, mirroring, or data accuracy verification. State policies the system must enforce for either retaining or disposing of data, including temporary data, metadata, residual data (such as deleted records), cached data, local copies, archives, and interim backups.]

Example:

DI-1: The COS shall retain individual Patron meal orders for 6 months following the meal's delivery date.

DI-2: The COS shall retain menus for 1 year following the menu date.

5. External Interface Requirements

[This section provides information to ensure that the system will communicate properly with users and with external hardware or software elements.]

5.1 User Interfaces

[Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.]

Example:

UI-1: The Cafeteria Ordering System screen displays shall conform to the Process Impact Internet Application User Interface Standard, Version 2.0 [3].
UI-2: The system shall provide a help link from each displayed webpage to explain how to use that page.
UI-3: The webpages shall permit complete navigation and food item selection by using the keyboard alone, in addition to using mouse and keyboard combinations.

5.2 Software Interfaces

[Describe the connections between this product and other software components (identified by name and version), including other applications, databases, operating systems, tools, libraries, websites, and integrated commercial components. State the purpose, formats, and contents of the messages, data, and control values exchanged between the software components. Specify the mappings of input and output data between the systems and any translations that need to be made for the data to get from one system to the other. Describe the services needed by or from external software components and the nature of the intercomponent communications. Identify data that will be exchanged between or shared across software components. Specify nonfunctional requirements affecting the interface, such as service levels for responses times and frequencies, or security controls and restrictions.]

Example:

SI-1: Cafeteria Inventory System

> *SI-1.1: The COS shall transmit the quantities of food items ordered to the Cafeteria Inventory System through a programmatic interface.*
>
> *SI-1.2: The COS shall poll the Cafeteria Inventory System to determine whether a requested food item is available.*
>
> *SI-1.3: When the Cafeteria Inventory System notifies the COS that a specific food item is no longer available, the COS shall remove that food item from the menu for the current date.*

SI-2: Payroll System

> *The COS shall communicate with the Payroll System through a programmatic interface for the following operations:*
>
> *SI-2.1: To allow a Patron to register and unregister for payroll deduction.*
>
> *SI-2.2: To inquire whether a Patron is registered for payroll deduction.*
>
> *SI-2.3: To inquire whether a Patron is eligible to register for payroll deduction.*
>
> *SI-2.4: To submit a payment request for a purchased meal.*

SI-2.5: To reverse a previous charge because a patron rejected a meal or wasn't satisfied with it, or because the meal was not delivered per the delivery instructions.

5.3 Hardware Interfaces

[Describe the characteristics of each interface between the software and hardware (if any) components of the system. This description might include the supported device types, the data and control interactions between the software and the hardware, and the communication protocols to be used. List the inputs and outputs, their formats, their valid values or ranges, and any timing issues developers need to be aware of. If this information is extensive, consider creating a separate interface specification document.]

Example:

No hardware interfaces have been identified.

5.4 Communications Interfaces

[State the requirements for any communication functions the product will use, including e-mail, Web browser, network protocols, and electronic forms. Define any pertinent message formatting. Specify communication security or encryption issues, data transfer rates, handshaking, and synchronization mechanisms. State any constraints around these interfaces, such as whether e-mail attachments are acceptable or not.]

Example:

CI-1: The COS shall send an email or text message (based on user account settings) to the Patron to confirm acceptance of an order, price, and delivery instructions.

CI-2: The COS shall send an email or text message (based on user account settings) to the Patron to report any problems with a meal order or delivery.

6. Quality Attributes

6.1 Usability

[Specify any requirements regarding characteristics that will make the software appear to be "user-friendly." Usability encompasses ease of use, ease of learning; memorability; error avoidance, handling, and recovery; efficiency of interactions; accessibility; and ergonomics. Sometimes these can conflict with each other, as with ease of use over ease of learning. Indicate any user interface design standards or guidelines to which the application must conform.]

Example:

USE-1: The COS shall allow a Patron to retrieve the previous meal ordered with a single interaction.

USE-2: 95% of new users shall be able to successfully order a meal without errors on their first try.

6.2 Performance

[State specific performance requirements for various system operations. If different functional requirements or features have different performance requirements, it's appropriate to specify those performance goals right with the corresponding functional requirements, rather than collecting them in this section.]

Example:

PER-1: The system shall accommodate a total of 400 users and a maximum of 100 concurrent users during the peak usage time window of 9:00 A.M. to 10:00 A.M. local time, with an estimated average session duration of 8 minutes.

PER-2: 95% of webpages generated by the COS shall download completely within 4 seconds from the time the user requests the page over a 20 Mbps or faster Internet connection.

PER-3: The system shall display confirmation messages to users within an average of 3 seconds and a maximum of 6 seconds after the user submits information to the system.

6.3 Security

[Specify any requirements regarding security or privacy issues that restrict access to or use of the product. These could refer to physical, data, or software security. Security requirements often originate in business rules, so identify any security or privacy policies or regulations to which the product must conform. If these are documented in a business rules repository, just refer to them.]

Example:

SEC-1: All network transactions that involve financial information or personally identifiable information shall be encrypted per BR-33.

SEC-2: Users shall be required to log on to the COS for all operations except viewing a menu.

SEC-3: Only authorized Menu Managers shall be permitted to work with menus, per BR-24.

SEC-4: The system shall permit Patrons to view only orders that they placed.

6.4 Safety

[Specify requirements that are concerned with possible loss, damage, or harm that could result from use of the product. Define any safeguards or actions that must be taken, as well as potentially dangerous actions that must be prevented. Identify any safety certifications, policies, or regulations to which the product must conform.]

Example:

SAF-1: The user shall be able to see a list of all ingredients in any menu items, with ingredients highlighted that are known to cause allergic reactions in more than 0.5 percent of the North American population.

6.5 Availability

Example:

AVL-1: The COS shall be available at least 98% of the time between 5:00 A.M. and midnight local time and at least 90% of the time between midnight and 5:00 A.M. local time, excluding scheduled maintenance windows.

6.6 Robustness

Example:

ROB-1: If the connection between the user and the COS is broken prior to a new order being either confirmed or terminated, the COS shall enable the user to recover an incomplete order and continue working on it.

6.7 [Others as relevant]

[Create a separate section in the SRS for each additional product quality attribute to describe characteristics that will be important to either customers or developers. Possibilities include availability, efficiency, installability, integrity, interoperability, modifiability, portability, reliability, reusability, robustness, scalability, and verifiability. Write these to be specific, quantitative, and verifiable. Clarify the relative priorities for various attributes, such as security over performance.]

7. Internationalization and Localization Requirements

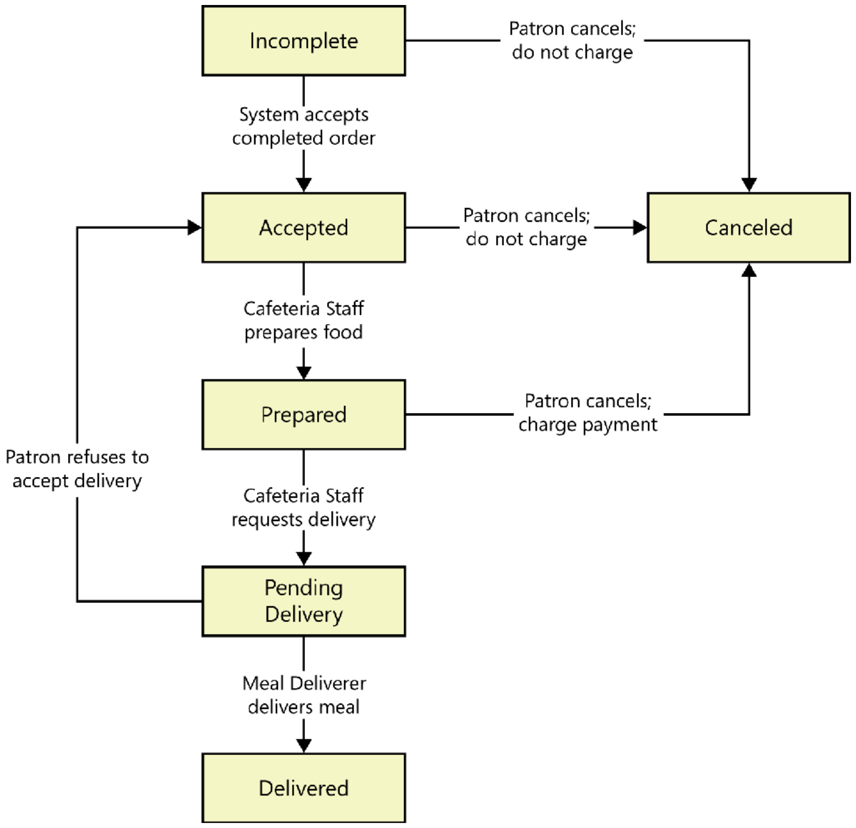
[Internationalization and localization requirements ensure that the product will be suitable for use in nations, cultures, and geographic locations other than those in which it was created. Such requirements might address differences in: currency; formatting of dates, numbers, addresses, and telephone numbers; language, including national spelling conventions within the same language (such as American versus British English), symbols used, and character sets; given name and family name order; time zones; international regulations and laws; cultural and political issues; paper sizes used; weights and measures; electrical voltages and plug shapes; and many others.]

8. Other Requirements

[Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections.]

9. Appendix A:

Figure C-4 is a state-transition diagram that shows the possible meal order statuses and the allowed changes in status.


Incomplete
Patron cancels; do not charge
System accepts completed order
Accepted
Patron cancels; do not charge
Canceled
Cafeteria Staff prepares food
Prepared
Patron cancels; charge payment
Patron refuses to accept delivery
Cafeteria Staff requests delivery
Pending Delivery
Meal Deliverer delivers meal
Delivered