

System Requirement Specification [Tile of Project]

[Current version]

[Supervisor]

[Students]

Hanoi, [dd-mm-yyyy]

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Project Scope and Product Features 1

1.4 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 User Classes and Characteristics 3

2.3 Operating Environment 3

2.4 Design and Implementation Constraints 4

2.5 Assumptions and Dependencies 4

3. System Features 4

3.1 Description 4

3.2 Functional requirements 5

4. Data Requirements 6

4.1 Logical Data Model 6

4.2 Data Dictionary 7

4.3 Reports 10

Example: Ordered Meal History Report 10

4.4 Data Integrity, Retention, and Disposal 11

5. External Interface Requirements 11

5.1 User Interfaces 11

5.2 Software Interfaces 11

5.3 Hardware Interfaces 12

5.4 Communications Interfaces 12

6. Quality Attributes 12

6.1 Usability Requirements 12

6.2 Performance Requirements 12

6.3 Security Requirements 12

6.4 Safety Requirements 13

6.5 Availability Requirements 13

6.6 Robustness Requirements 13

Appendix A: Analysis Models 14

# Introduction

*[The introduction presents an overview to help the reader understand how the SRS is organized and how to use it.]*

## Purpose

*[Identify the product or application whose requirements are specified in this document, including the revision or release number. If this SRS pertains to only part of a complex system, identify that portion or subsystem. Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers.]*

<<sample This SRS describes the functional and nonfunctional requirements for software release 1.0 of the Cafeteria Ordering System (COS). 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.>>

## Document Conventions

*[Describe any standards or typographical conventions used, including the meaning of specific text styles, highlighting, or notations. If you are manually labeling requirements, you might specify the format here for anyone who needs to add one later.]*

## Project Scope and Product Features

*[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.]*

<<sample The COS will permit Process Impact employees to order meals from the company cafeteria on-line 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.>>

## 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, source, storage location, or URL.]*

<<sample

1. Wiegers, Karl. *Cafeteria Ordering System Vision and Scope Document*, www.processimpact.com/projects/COS/COS Vision and Scope.docx
2. Beatty, Joy. *Process Impact Intranet Development Standard, Version 1.3*, www.processimpact.com/corporate/standards/PI Intranet Development Standard.pdf
3. Rath, Andrew. *Process Impact Internet Application User Interface Standard, Version 2.0*, www.processimpact.com/corporate/standards/PI Internet UI Standard.pdf
4. Wiegers, Karl. *Software Requirements 3rd Edition,*

>>>

# Overall Description

*[This section presents a high-level overview of the product and the environment in which it will be used, the anticipated users, and known constraints, assumptions, and dependencies.]*

## 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 or ecosystem map (described in Chapter 5) to show the product’s relationship to other systems.]*

<<sample 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 1 illustrates the external entities and system interfaces for release 1.0. 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.



Figure 1. Context diagram for release 1.0 of the Cafeteria Ordering System.

>>

## User Classes and Characteristics

*[Identify the various user classes that you anticipate will use this product, and describe their pertinent characteristics. (See Chapter 6, “Finding the voice of the user.”) 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 a master user class catalog is available, you can incorporate user class descriptions by simply pointing to them in the catalog instead of duplicating information here.]*

<<sample

|  |  |
| --- | --- |
| 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. |

## 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.]*

<<sample

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, iOS, and Windows smartphones and tablets.

>>

## Design and Implementation Constraints

*[There are times when a certain programming language must be used, a particular code library that has already had time invested to develop it needs to be used, and so forth. Describe any factors that will restrict the options available to the developers and the rationale for each constraint. Requirements that incorporate or are written in the form of solution ideas rather than needs are imposing design constraints, often unnecessarily, so watch out for those. ]*

<<sample

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.

>>

## Assumptions and Dependencies

*[An assumption is a statement that is believed to be true in the absence of proof or definitive knowledge. Problems can arise if assumptions are incorrect, are obsolete, are not shared, or change, so certain assumptions will translate into project risks. One SRS reader might assume that the product will conform to a particular user interface convention, whereas another might assume something different. A developer might assume that a certain set of functions will be custom-written for this application, whereas the business analyst might assume that they will be reused from a previous project, and the project manager might expect to procure a commercial function library.]*

<<sample

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.

>>

# System Features

## Description

*[Provide a short description of the feature and indicate whether it is of high, medium, or low priority. (See Chapter 16, “First things first: Setting requirement priorities.”) Priorities often are dynamic, changing over the course of the project. If you’re using a requirements management tool, define a requirement attribute for priority.]*

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

## Functional requirements

*[Itemize the specific functional requirements associated with this 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. Describe how the product should respond to anticipated error conditions and to invalid inputs and actions. Uniquely label each functional requirement, as described earlier in this chapter. If you’re using a requirements management tool, you can create multiple attributes for each functional requirement, such as rationale, origin, and status.]*

<<sample

|  |
| --- |
| **Order.Place: Placing a meal order**  .Register: The COS shall confirm that the Patron is registered for payroll deduction.  .No: If the Patron is not registered for payroll deduction, the COS shall give the Patron options to register now and continue placing an order, to place an order for pickup in the cafeteria (but not for delivery), or to exit.  .Date: The COS shall prompt the Patron for the meal date (see BR-8).  .Cutoff: If the meal date is the current date and the current time is after the order cutoff time, the COS shall inform the patron that it’s too late to place an order for today. The Patron can either change the meal date or cancel the order. |
| **Order.Deliver: Delivery or pickup**  .Select: The Patron shall specify whether the order is to be picked up or delivered.  .Location: If the order is to be delivered and there are still available delivery times for the meal date, the Patron shall provide a valid delivery location.  .Notimes: The COS shall notify the Patron if there are no available delivery times for the meal date. The Patron shall either cancel the order or indicate that he will pick up the order in the cafeteria.  .Times: The COS shall display the remaining available delivery times for the meal date. The COS shall allow the Patron to request one of the delivery times shown, to change the order to be picked up in the cafeteria, or to cancel the order. |
| **Order.Menu: Viewing a menu**  .Date: The COS shall display a menu for the date that the Patron specified.  .Available: The menu for the specified date shall display only those food items for which at least one unit is available in the cafeteria’s inventory and which may be delivered. |
| **Order.Units: Ordering multiple meals and multiple food items**  .Multiple: The COS shall permit the user to order multiple identical meals, up to the fewest available units of any menu item in the order.  .TooMany: If the Patron orders more units of a menu 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.Confirm: Confirming an order**  .Display: When the Patron indicates that he does not wish to order any more food items, the COS shall display the food items ordered, the individual food item prices, and the payment amount calculated per BR-12.  .Prompt: The COS shall prompt the Patron to confirm the meal order.  .Response: The Patron can confirm, edit, or cancel the order.  .More: The COS shall let the Patron order additional meals for the same or for a different date. BR-3 and BR-4 pertain to multiple meals in a single order. |
| **Order.Pay: Meal order payment**  .Method: When the Patron indicates that he is done placing orders, the COS shall ask the user to select a payment method.  .Deliver: See BR-11.  .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.  .Deduct: If the Patron selected payroll deduction, the COS shall issue a payment request to the Payroll System.  .OK: If the payment request is accepted, the COS shall display a message confirming acceptance of the order with a transaction number.  .NG: 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.Done: When the Patron has confirmed the order, the COS shall do the following as a single transaction.**  .Store: Assign the next available meal order number to the meal and store the meal order with a status of “Accepted.”  .Inventory: Send a message to the Cafeteria Inventory System with the number of units of each food item in the order.  .Menu: Update the menu for the current order’s order date to reflect any items that are now out of stock in the cafeteria inventory.  .Times: Update the remaining available delivery times for the date of this order.  .Patron: Send an email message or text message to the Patron with the meal order and meal payment information.  .Cafeteria: Send an email message to the Cafeteria Staff with the meal order information.  .Failure: 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. |

[Note: Functional requirements for reordering a meal and for changing and canceling meal orders are not provided in this example.]

# Data Requirements

*[Information systems provide value by manipulating data. Use this section of the template to describe various aspects of the data that the system will consume as inputs, process in some fashion, or create as outputs]*

## Logical Data Model

*[As described in Chapter 13, a data model is a visual representation of the data objects and collections the system will process and the relationships between them. Numerous notations exist for data modeling, including entity-relationship diagrams and UML class diagrams. You might include a data model for the business operations being addressed by the system, or a logical representation for the data that the system will manipulate. This is not the same thing as an implementation data model that will be realized in the form of database design.]*

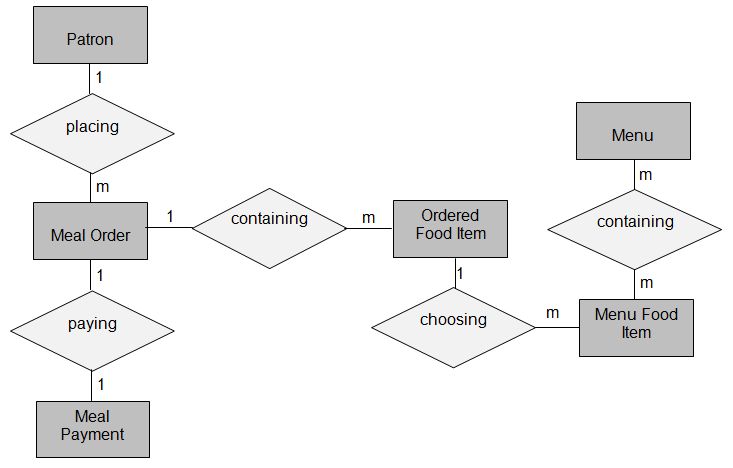


Figure . Partial data model for release 1.0 of the Cafeteria Ordering System.

## 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. Commercial data modeling tools often include a data dictionary component. 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.]*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | ext 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 |  |
| 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 |  |
| order date | the date on which a patron placed a meal order | date, MM/DD/YYYY | 10 |  |
| ordered food item | one menu food item that a Patron requested as part of a meal order | menu food item  + quantity ordered |  |  |
| patron | a Process Impact employee who is authorized to order a meal | patron name  + employee ID  + patron phone number  + patron location  + patron email |  |  |
| patron email | email address of the employee who placed a meal order | alphanumeric | 50 |  |
| patron location | building and room numbers of the employee who placed a meal order | alphanumeric | 50 | hyphens and commas permitted |
| patron name | name of the employee who placed a meal order | alphabetic | 30 |  |
| patron phone number | telephone number of the employee who placed a meal order | AAA-EEE-NNNN xXXXX for area code (A), exchange (E), number (N), and extension (X) | 18 |  |
| payment amount | total price of an order in dollars and cents, calculated per BR-12 | numeric, dollars and cents | dddd.cc |  |
| payment method | how the Patron is paying for a meal he ordered | alphabetic | 16 | payroll deduction, cash, credit card, debit card |
| quantity ordered | the number of units of each food item that the Patron is ordering in a single meal order | integer | 4 | default = 1; maximum = quantity presently in inventory |
| transaction number | unique sequence number that COS assigns to each payment transaction | integer | 12 |  |

## Reports

**Example: Ordered Meal History Report**

|  |  |
| --- | --- |
| 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 six 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 |

[Note: Other COS reports are not provided in this example.]

[*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.*]

## Data Integrity, Retention, and Disposal

[If relevant, describe how data is acquired and maintained. For instance, when starting a data inventory feed, you might need to do an initial dump of all the inventory data to the receiving system and then have subsequent feeds that consist only of changes. 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]

<<sample

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 one year following the menu date.

>>

# External Interface Requirements

## User Interfaces

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.

## Software Interfaces

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 all or part of a previous charge because a patron rejected a meal or wasn’t satisfied with it, or because the meal was not delivered per the confirmed delivery instructions.

## Hardware Interfaces

No hardware interfaces have been identified.

## Communications Interfaces

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 the meal order or delivery.

# Quality Attributes

## Usability Requirements

*[Usability requirements deal with ease of learning, ease of use, error avoidance and recovery, efficiency of interactions, and accessibility. The usability requirements specified here will help the user interface designer create the optimum user experience.]*

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.

## Performance Requirements

*[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.]*

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 20Mbps 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.

## Security Requirements

*[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]*

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.

## Safety Requirements

*[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.]*

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.

## Availability Requirements

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.

## Robustness Requirements

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.

# Appendix A: Analysis Models

Figure 3 is a state-transition diagram that shows the possible meal order statuses and the allowed changes in status.

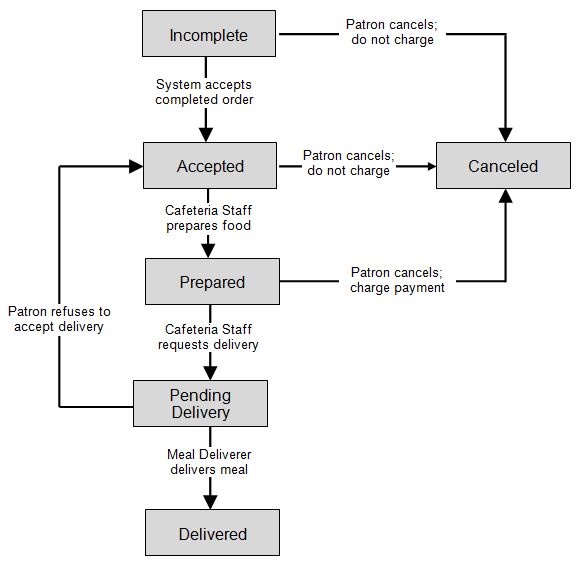
****

Figure 3. State-transition diagram for meal order status.

*[Define any specialized terms that a reader needs to know to understand the SRS, including acronyms and abbreviations. Spell out each acronym and provide its definition. Consider building a reusable enterprise-level glossary that spans multiple projects and incorporating by reference any terms that pertain to this project.]*