**Inflectra Corporation**

**8121 Georgia Ave, Suite 504**

**Silver Spring. MD 20910-4957**

**+1 202.558.6885**

**www.inflectra.com**

**Inflectra Corporation |** Sample MS-Word Import Template

*By [Author]*

*[Role]*

Contents

[1. Executive Summary 1](#_Toc255847870)

[2. Functional Requirements 2](#_Toc255847871)

[2.1. Context 2](#_Toc255847872)

[2.2. Business Features 2](#_Toc255847873)

[2.3. Use Cases 2](#_Toc255847874)

[2.4. Data Flow Diagrams 4](#_Toc255847875)

[3. Other Requirements 4](#_Toc255847876)

[3.1. Interface Requirements 4](#_Toc255847877)

[3.2. Data Conversion Requirements 5](#_Toc255847878)

[3.3. Hardware/Software Requirements 5](#_Toc255847879)

[3.4. Operational Requirements 5](#_Toc255847880)

[4. Functional Test Cases 8](#_Toc255847881)

[4.1. Test Case 1 8](#_Toc255847882)

[4.2. Test Case 2 8](#_Toc255847883)

[5. Regression Test Cases 9](#_Toc255847884)

[5.1. Test Case 1 9](#_Toc255847885)

[5.2. Test Case 2 9](#_Toc255847886)

# 1. Executive Summary

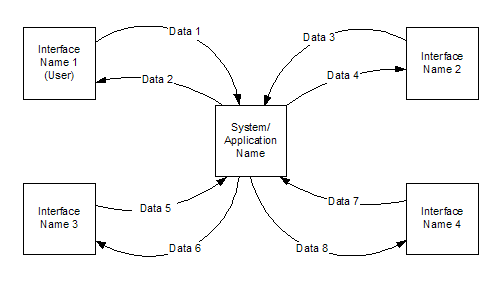
Introductory text goes here.

This is a summary of the main features of the application.

# 2. Functional Requirements

## 2.1. Context

[Provide a context diagram of the system, with explanations as applicable. The context of a system refers to the connections and relationships between the system and its environment.]



## 2.2. Business Features

[This section describes all the business features of the system]

### 2.2.1. Feature 1

[The system shall do X]

### 2.2.2. Feature 2

[The system shall do X]

#### 2.2.2.1. Feature 2.1

[The system shall do X]

#### 2.2.2.2. Feature 2.2

[The system shall do X]

### 2.2.3. Feature 3

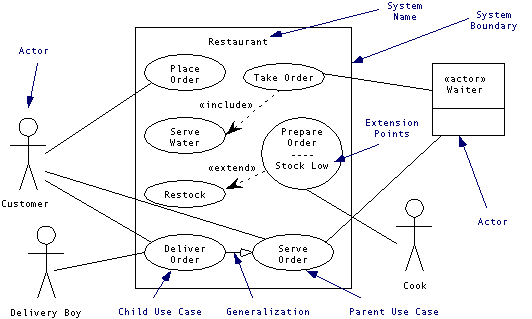
[The system shall do X]

## 2.3. Use Cases

[This section contains all the use-cases / scenarios that illustrate the business features in context]

### 2.3.1. Use Case 1

The following use-diagram illustrates the functionality:

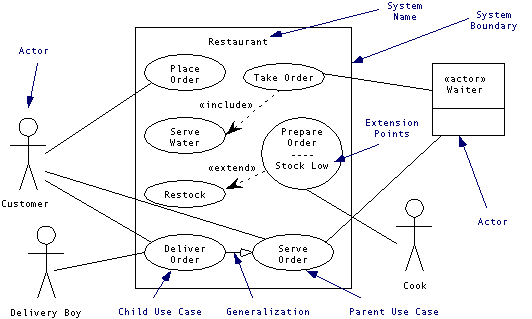


These steps are illustrated with the following scenario:

|  |  |  |
| --- | --- | --- |
| **Step Number** | **Actor** | **Action** |
| 1 | Customer | Places Order |
| 2 | Restaurant | Prepares Order |
| 3 | Restaurant | Delivers Order |
| 4 | Customer | Pays for Order |

### 2.3.2. Use Case 2

The following use-diagram illustrates the functionality:



#### 2.3.2.1. Use Case 2.1

TBD

#### 2.3.2.2. Use Case 2.2

TBD

### 2.3.3. Use Case 3

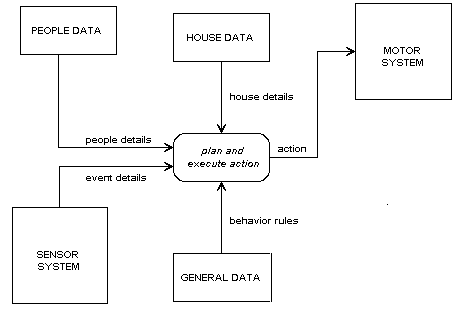
TBD

## 2.4. Data Flow Diagrams

[This section contains any data flow diagrams that illustrate how data will flow in the system]

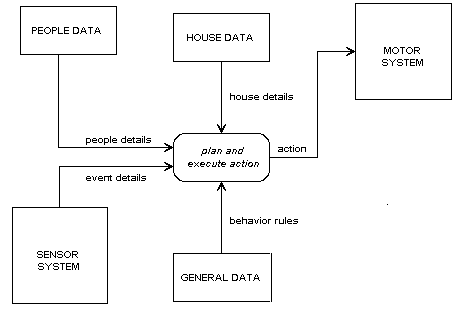
### 2.4.1. Data Flow Diagram 1

The following use-diagram illustrates the functionality:



### 2.4.2. Data Flow Diagram 2

The following use-diagram illustrates the functionality:



# 3. Other Requirements

[Describe the non-behavioral requirements.]

## 3.1. Interface Requirements

[Describe the user interfaces that are to be implemented by the system.]

### 3.1.1. Hardware Interfaces

[Define hardware interfaces supported by the system, including logical structure, physical addresses, and expected behavior.]

### Software Interfaces

[Name the applications with which the subject application must interface. State the following for each such application: name of application, external owner of application, interface details (only if determined by the other application).

It is acceptable to reference an interface control document for details of the interface interactions.]

### Communications Interfaces

[Describe communications interfaces to other systems or devices, such as local area networks.]

## 3.2. Data Conversion Requirements

[Describe the requirements needed for conversion of legacy data into the system.]

## 3.3. Hardware/Software Requirements

[Provide a description of the hardware and software platforms needed to support the system.]

## 3.4. Operational Requirements

[Provide the operational requirements in this section.

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be”? Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs, preferences are not. If, for example, the user requires a special response but does not have a business-related reason for it, that requirement is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below.]

Operational requirements describe how the system will run and communicate with operations personnel.

### 3.4.1. Security and Privacy

[Provide a list of the security requirements using the following criteria:

A. State the consequences of the following breaches of security in the subject application:

1. Loss or corruption of data
2. Disclosure of secrets or sensitive information
3. Disclosure of privileged/privacy information about individuals
4. Corruption of software or introduction of malware, such as viruses
5. State the type(s) of security required. Include the need for the following as appropriate:
6. Physical security.
7. Access by user role or types.
8. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.
9. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only".
10. State if there is a need for certification and accreditation of the security measures adopted for this application]

*The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data.*

### 3.4.2. Audit Trail

[List the activities recorded in the application’s audit trail. For each activity, list the data recorded.]

### 3.4.3. Reliability

A. [State the following in this section:

1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
2. Loss of human life
3. Complete or partial loss of the ability to perform a mission-critical function
4. Loss of revenue
5. Loss of employee productivity
6. What is the minimum acceptable level of reliability?

B. State required reliability:

1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]

*Reliability is the probability that the system processes work correctly and completely without being aborted.*

### 3.4.4. Recoverability

[Answer the following questions in this section:

A. In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?

B. In the event the database is corrupted, to what level of currency must it be restored? For example “The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred”.

C. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

*Recoverability is the ability to restore function and data in the event of a failure.*

### 3.4.5. System Availability

[State the period during which the application must be available to users. For example, “*The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST.* If the application must be available to users in more than one time zone, state the earliest start time and the latest stop time. Consider daylight savings time, too.

Include use peak times. These are times when system unavailability is least acceptable.]

*System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.*

### 3.4.6. General Performance

[Describe the requirements for the following:

A. Response time for queries and updates

B. Throughput

C. Expected rate of user activity (for example, number of transactions per hour, day, or month, or cyclical periods)

Specific performance requirements, related to a specific functional requirement, should be listed with that functional requirement.

### 3.4.7. Capacity

[List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide them]

### 3.4.8. Data Retention

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]

### 3.4.9. Error Handling

[Describe system error handling.]

### 3.4.10. Validation Rules

[Describe System Validation Rules.]

### 3.4.11. Conventions/Standards

[Describe system conventions and standards followed.

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

# 4. Functional Test Cases

Functional tests confirm the agreed upon capabilities of the implemented solution and are based on the Use Cases defined in the requirements specification. Functional tests are specifically designed to validate that the Solution addresses a requirement as closely as possible. The functional test for this project will be conducted over a <*insert time period*> once the implementation is complete, but prior to deployment of the Solution into production (see the updated Project Schedule).

Test cases will step through the scenarios listed in the Use Cases defined in the requirements specification. Test scripts, based on the Test Case definitions below, will prescribe a set of steps that each actor will take during the test. A place for the tester to indicate whether expected results were met and to document comments will also be provided.

## 4.1. Test Case 1

**Brief Description:**

(Can come from the Use Case document.

**Flow of Events:**

(Comes from the Flow of Events from the Use Case document.)

**Test Entry Criteria:**

(Comes from the Pre-Condition section in the Use Case document.

**Test Exit Criteria:**

(Comes from the Use Case document, section - Post Conditions)

**Test Steps:**

The following test steps should be completed by the tester:

|  |  |  |
| --- | --- | --- |
| **Description** | **Expected Result** | **Sample Data** |
| User clicks on button to do X | The system does Y |  |
| User enters some test into box | The system does nothing | Some Text |
| User clicks button to submit form | The system displays confirmation screen. |  |

## 4.2. Test Case 2

**Brief Description:**

(Can come from the Use Case document.

**Flow of Events:**

(Comes from the Flow of Events from the Use Case document.)

**Test Entry Criteria:**

(Comes from the Pre-Condition section in the Use Case document.

**Test Exit Criteria:**

(Comes from the Use Case document, section - Post Conditions)

**Test Steps:**

The following test steps should be completed by the tester:

|  |  |  |
| --- | --- | --- |
| **Description** | **Expected Result** | **Sample Data** |
| User clicks on button to do X | The system does Y |  |
| User enters some test into box | The system does nothing | Some Text |
| User clicks button to submit form | The system displays confirmation screen. |  |

# 5. Regression Test Cases

Regression testing should be carried out on any parts of the system that have been upgraded or patched during the lifetime of the release. An appropriate regression test would involve appropriate unit and integration testing related to the affected module; and a specific test or set of tests related to any functional or non-functional test that involves the use of the upgraded or patched software.

The regression testing should not be so extensive that an entire execution of the functional tests is required. The regression test should only focus on the function specifically provided by the upgraded or patched component of the software product.

## 5.1. Test Case 1

**Brief Description:**

(Can come from the Use Case document.

**Flow of Events:**

(Comes from the Flow of Events from the Use Case document.)

**Test Entry Criteria:**

(Comes from the Pre-Condition section in the Use Case document.

**Test Exit Criteria:**

(Comes from the Use Case document, section - Post Conditions)

**Test Steps:**

The following test steps should be completed by the tester:

|  |  |  |
| --- | --- | --- |
| **Description** | **Expected Result** | **Sample Data** |
| User clicks on button to do X | The system does Y |  |
| User enters some test into box | The system does nothing | Some Text |
| User clicks button to submit form | The system displays confirmation screen. |  |

## 5.2. Test Case 2

**Brief Description:**

(Can come from the Use Case document.

**Flow of Events:**

(Comes from the Flow of Events from the Use Case document.)

**Test Entry Criteria:**

(Comes from the Pre-Condition section in the Use Case document.

**Test Exit Criteria:**

(Comes from the Use Case document, section - Post Conditions)

**Test Steps:**

The following test steps should be completed by the tester:

|  |  |  |
| --- | --- | --- |
| **Description** | **Expected Result** | **Sample Data** |
| User clicks on button to do X | The system does Y |  |
| User enters some test into box | The system does nothing | Some Text |
| User clicks button to submit form | The system displays confirmation screen. |  |