# Functional Modeling

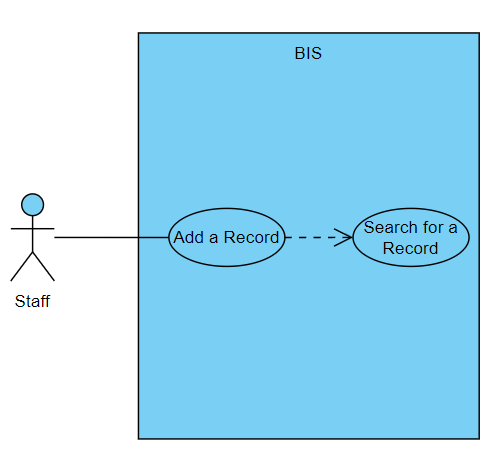
This section of the document will model the functional requirements of the Book Inventory System (BIS) based on the following use cases:

* Adding a Book or Order Record
* Adding a Customer or Employee Record
* Removing a Record
* Adjusting a Record
* Listing All Records of a Given Type
* Searching for a Record

## Adding a Book or Order Record

Functional Requirement Modeled: FR1 – Manipulate a Record

The following use case diagram displays the process of adding a book or order record to the BIS. These two types of records differ from the others because they both employ the “Search for a Record” process to verify the presence of previously existing records involved in the process.



### Adding a Book Record

The following procedure describes the creation of a new book record. The key data member for this kind of record is the ISBN. The ISBN can only contain numbers, and it must be exactly 10 or 13 digits long based on the standards established for ISBNs by the authority organization, the International ISBN Agency. Each book record must include a non-negative, numerical Store Price for the book.

#### Normal Scenario

1. Staff enters the new record’s key data member, the ISBN.
2. BIS validates the ISBN against the expected format.
3. BIS searches the existing book records by ISBN to verify that the ISBN is not already in use.
4. Staff enters information for all of the book’s remaining mandatory data members: Title, at least one Author, Format, Edition, Store Price, and Stock quantity.
5. BIS validates the Store Price against the expected format.
6. BIS creates a new book record with this information.

#### Exception Scenario – Key Data Member Conflicts with An Existing Record

1. Staff enters the new record’s key data member, the ISBN.
2. BIS validates the ISBN against the expected format.
3. BIS searches the existing book records and discovers one that already uses that ISBN.
4. BIS displays the existing book record with a message informing the user that the ISBN provided is already in use.

#### Exception Scenario – Invalid Data for Key Data Member

1. Staff enters the new record’s key data member, the ISBN.
2. BIS fails to validate the ISBN against the expected format.
3. BIS displays an error message informing the user that the ISBN does not meet validation requirements in terms of either limits or content as appropriate.

#### Exception Scenario – Invalid Data for Store Price

1. Staff enters the new record’s key data member, the ISBN.
2. BIS validates the ISBN against the expected format.
3. BIS searches the existing book records by ISBN to verify that the ISBN is not already in use.
4. Staff enters information for all of the book’s remaining mandatory data members: Title, at least one Author, Format, Edition, Store Price, and Stock quantity.
5. BIS fails to validate the Store Price against the expected format.
6. BIS displays an error message informing the user that the Store Price does not meet validation requirements in terms of a non-zero value.

### Adding an Order Record

Functional Requirement Modeled: FR1 – Manipulate a Record

This procedure describes the creation of a new order record. The key data member for this record is the Order Number, which contains only numbers and is ten digits in length. Each record must also provide a valid Employee Number and at least one book. Books in orders must include an ISBN and an Order Quantity above 0.

#### Normal Scenario

1. BIS auto-fills the new Order Number, one higher than the most recent prior Order Number entered.
2. Staff enters their assigned Employee Number.
3. BIS validates the Employee Number against the expected format.
4. BIS searches employee records for one with a matching Employee Number.
5. Staff enters the information for the order’s Contents, including at least one ISBN with a non-zero quantity.
6. BIS validates each ISBN and order quantity in the order’s Contents against expected formats.
7. BIS searches the book records by ISBN to verify that each book exists in the system and that there is a sufficient Stock quantity to satisfy the order.
8. Staff enters the information for the order’s optional data member, the Customer Number.
9. BIS validates the Customer Number against the expected format as applicable.
10. BIS searches the customer records by Customer Number to verify a customer with that number exists in the system.
11. BIS creates a new order record with this information.
12. BIS decreases the Stock quantity for each book sold as part of the order Contents accordingly.

#### Exception Scenario – Invalid Employee Number

1. BIS auto-fills the new Order Number, one higher than the most recent prior Order Number entered.
2. Staff enters their assigned Employee Number.
3. BIS fails to validate the Employee Number against the expected format or the employee list.
4. BIS exits the process while displaying an error message explaining the failure.

#### Exception Scenario – Invalid ISBN for Book in Order Contents

1. BIS auto-fills the new Order Number, one higher than the most recent prior Order Number entered.
2. Staff enters their assigned Employee Number.
3. BIS validates the Employee Number against the expected format.
4. BIS searches employee records for one with a matching Employee Number.
5. Staff enters the information for the order’s Contents, including at least one ISBN with a non-zero quantity.
6. BIS fails to validate at least one ISBN in the Order Contents.
7. BIS exits the process while displaying an error message explaining the failure.

#### Exception Scenario – Insufficient Stock Quantity for Book in Order Contents

1. BIS auto-fills the new Order Number, one higher than the most recent prior Order Number entered.
2. Staff enters their assigned Employee Number.
3. BIS validates the Employee Number against the expected format.
4. BIS searches employee records for one with a matching Employee Number.
5. Staff enters the information for the order’s Contents, including at least one ISBN with a non-zero quantity.
6. BIS discovers at least one book in the Order Contents has insufficient stock quantity to cover the desired order quantity.
7. BIS exits the process while displaying an error message explaining the failure.

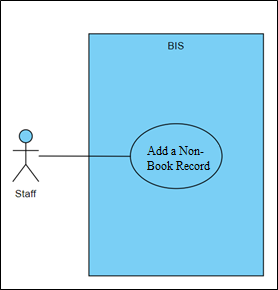
#### Exception Scenario – Invalid Customer Number

1. BIS auto-fills the new Order Number, one higher than the most recent prior Order Number entered.
2. Staff enters their assigned Employee Number.
3. BIS validates the Employee Number against the expected format.
4. BIS searches employee records for one with a matching Employee Number.
5. Staff enters the information for the order’s Contents, including at least one ISBN with a non-zero quantity.
6. BIS validates each ISBN and order quantity in the order’s Contents against expected formats.
7. BIS searches the book records by ISBN to verify that each book exists in the system and that there is a sufficient Stock quantity to satisfy the order.
8. Staff enters the information for the order’s optional data member, the Customer Number.
9. BIS fails to validate the Customer Number against the expected format or locate a matching Customer Number in the database.
10. BIS exits the process while displaying an error message describing the failure.

## Adding a Customer or Employee Record

Functional Requirement Modeled: FR1 – Manipulate a Record

The following use case describes the process of adding a customer or employee record to the BIS.



### Adding a Customer Record

This procedure describes the creation of a new customer record. The key data member for this kind of record is the Customer Number. This data member contains only numbers and is ten digits in length. It is assigned automatically by the system to preempt possible conflicts.

#### Normal Scenario

1. BIS auto-fills the new Employee Number, one higher than the most recent prior Employee Number entered.
2. Staff selects if the customer is an individual or a business, which determines whether the record requires a First Name and Last Name or a Business Name.
3. Staff enters information for the customer’s remaining mandatory data members: First/Last Name or Business Name, Mailing Address, Phone Number, and Email Address.
4. BIS creates a new customer record with this information.

### Adding an Employee Record

This procedure describes the creation of a new employee record. The key data member for this type of record is the Employee Number. This data member contains only numbers and is ten digits in length.

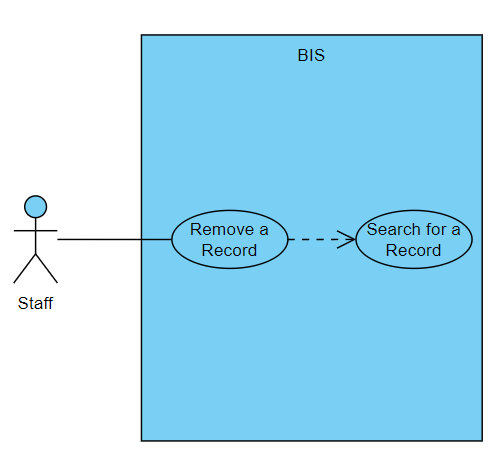
#### Normal Scenario

1. BIS auto-fills the new Employee Number, one higher than the most recent prior Employee Number entered.
2. Staff enters information for the employee’s remaining mandatory data members: First Name, Last Name, Mailing Address, Phone Number, and Email Address.
3. BIS creates a new employee record with this information.

## Removing a Record

Functional Requirement Modeled: FR1 – Manipulate a Record

The following use case diagram displays the process of removing a record from the BIS.



This procedure describes the removal of an existing record from the system.

#### Normal Scenario

1. Staff inputs a key data member for the record of their choice.
2. BIS validates the key data against the expected format.
3. BIS searches the database for a record of the matching type with a matching key data member.
4. BIS removes that record from the database while displaying a confirmation message.

#### Exception Scenario – Invalid Key Data Member

1. Staff inputs a key data member for the record of their choice.
2. BIS fails to validate the key data member against the expected format.
3. BIS exits the process while displaying an error message describing the failure.

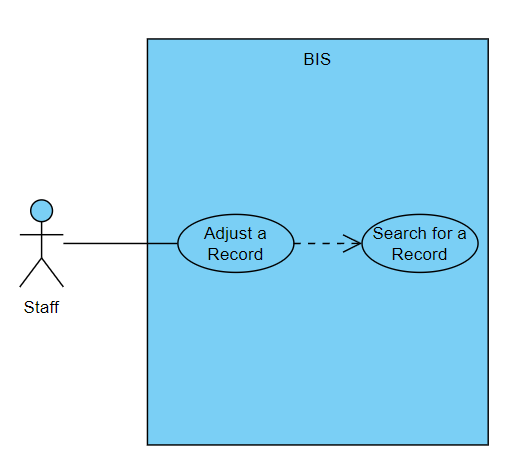
#### Exception Scenario – Matching Record Not Found

1. Staff inputs a key data member for the record of their choice.
2. BIS validates the key data member against the expected format.
3. BIS fails to locate a record with the matching key data member.
4. BIS exits the process while displaying an error message describing the failure.

## Adjusting a Record

Functional Requirement Modeled: FR1 – Manipulate a Record

The following use case diagram displays the process of adjusting a record already in the BIS. This procedure does not permit the alteration of a key data member.



This procedure describes the adjustment of any property of any existing record. The most frequent application of this use case involves adjusting the stock quantity or price of a book.

#### Normal Scenario – Record Found

1. Staff inputs a key data member for the record of their choice.
2. BIS validates the key data against the expected format.
3. BIS searches the database for a record of the matching type with a matching key data member.
4. BIS displays the record in a text readout.
5. Staff updates values in updatable fields.
6. BIS validates any altered fields against expected format, if applicable.
7. BIS updates the record while displaying a confirmation message.

#### Exception Scenario – Invalid Key Data Member

1. Staff inputs a key data member for the record of their choice.
2. BIS fails to validate the key data member against the expected format.
3. BIS exits the process while displaying an error message describing the failure.

#### Exception Scenario – Matching Record Not Found

1. Staff inputs a key data member for the record of their choice.
2. BIS validates the key data member against the expected format.
3. BIS fails to locate a record with the matching key data member.
4. BIS exits the process while displaying an error message describing the failure.

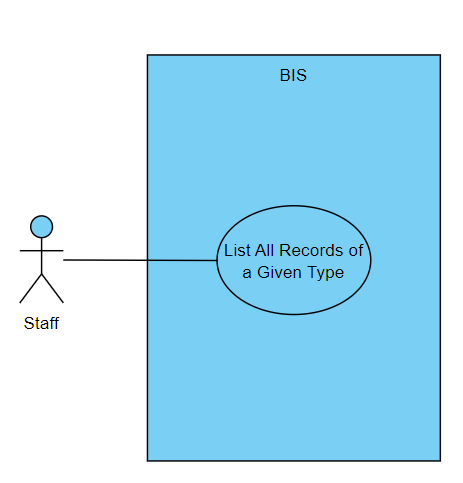
#### Exception Scenario – New Value is Invalid

1. Staff inputs a key data member for the record of their choice.
2. BIS validates the key data against the expected format.
3. BIS searches the database for a record of the matching type with a matching key data member.
4. BIS displays the record in a text readout.
5. Staff updates values in updatable fields.
6. BIS fails to validate updated data against expected format.
7. BIS exits the process while displaying an error message describing the failure, without updating the existing record.

## Listing All Records of a Given Type

Functional Requirement Modeled: FR2 – Print List of Records

This use case analyzes the generation of a report listing all records of a single type.



This procedure describes the listing of all records of a given type.

#### Normal Scenario

1. Staff selects a type of record from the following list: Book, Customer, Employee, Order.
2. BIS displays a list of all records of that type in the database.

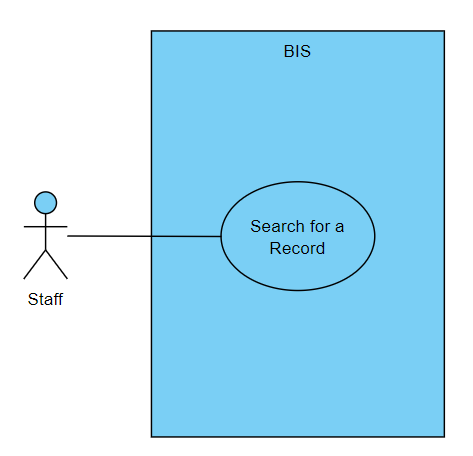
#### Exception Scenario – No Records Exist of Selected Type

1. Staff selects a type of record from the following list: Book, Customer, Employee, Order.
2. BIS displays an error message informing the user that there are no records of that type.

## Searching for a Record

Functional Requirement Modeled: FR3 – Search for a Record

The following use case diagram displays the process of searching for a record in the BIS.



This procedure describes the search for a record of a given type.

#### Normal Scenario – Record Found

1. Staff inputs a key data member for the record of their choice.
2. BIS validates the key data against the expected format.
3. BIS searches the database for a record of the matching type with a matching key data member.
4. BIS displays the record in a text readout.

#### Exception Scenario – Invalid Key Data Member

1. Staff inputs a key data member for the record of their choice.
2. BIS fails to validate the key data member against the expected format.
3. BIS exits the process while displaying an error message describing the failure.

#### Exception Scenario – Matching Record Not Found

1. Staff inputs a key data member for the record of their choice.
2. BIS validates the key data member against the expected format.
3. BIS fails to locate a record with the matching key data member.
4. BIS exits the process while displaying an error message describing the failure.

# Entity Modeling

This section of the document will model the components of the program and their data members. It includes classes for Book records, Customer records, Employee records, Order records, and the overall controller and input/output class that manages the GUI and all operations.

