Requirements Engineering

KennelSYS

Submitted By: Peter Mc Cafferty (T-00095451)

Computing with Software Development

Date Submitted: dd/mm/yyyy

**Table of Contents**

[1. Introduction/overview 4](#_Toc527042905)

[2. Functional Components 5](#_Toc527042906)

[3. User Requirements 6](#_Toc527042907)

[3.1. The KennelSYS will perform Kennel administration. 6](#_Toc527042908)

[3.2. KennelSYS will manage Bookings. 6](#_Toc527042909)

[3.3. KennelSYS will perform Administration. 6](#_Toc527042910)

[4. System Requirements 7](#_Toc527042911)

[System Level Use Case Diagram 7](#_Toc527042912)

[7](#_Toc527042913)

[7](#_Toc527042914)

[4.1 Manage Kennels 8](#_Toc527042915)

[4.1.1. Add Kennel Type 9](#_Toc527042916)

[4.1.2. Update a kennel Rate 12](#_Toc527042917)

[4.1.3. Remove a Kennel Type 15](#_Toc527042918)

[4.1.4 Kennel Enquiry 17](#_Toc527042919)

[4.2 Module 2 (Use active Verb + Noun) 19](#_Toc527042920)

[4.2.4 Functional Requirement 1 (Use active verb + noun) 19](#_Toc527042921)

[4.2.5 Functional Requirement 2 (Use active verb + noun) 19](#_Toc527042922)

[4.3 Module 3 (Use active Verb + Noun) 19](#_Toc527042923)

[4.3.4 Functional Requirement 1 (Use active verb + noun) 19](#_Toc527042924)

[4.3.5 Functional Requirement 2 (Use active verb + noun) 19](#_Toc527042925)

[5. System Model 20](#_Toc527042926)

[5.1. Level-0 DFD 20](#_Toc527042927)

[5.2. Level-1 DFD 20](#_Toc527042928)

[5.3. Level-2 DFD (Process P1: Title) 20](#_Toc527042929)

[5.4. Level-2 DFD (Process P2: Title) 20](#_Toc527042930)

[5.5. Level-2 DFD (Process P3: Title) 20](#_Toc527042931)

[6. Data Model (Class Diagram) 21](#_Toc527042932)

[6.1. Class Diagram 21](#_Toc527042933)

[6.2. Relational Schema 21](#_Toc527042934)

[6.3. Database Schema 21](#_Toc527042935)

[7. Conclusion 22](#_Toc527042936)

[8. Appendices 23](#_Toc527042937)

[8.1. Appendix A – Title 23](#_Toc527042938)

[8.2. Appendix B – Title 23](#_Toc527042939)

# Introduction/overview

The following software engineering project is called KennelSYS. It will provide functionality for the operation of a boarder kennel establishment. This software is for the use of those working in the boarder kennel. Some of the functionality provided by this system is:

* Manage Kennels
* Manage Bookings
* Perform Analysis

The management of the kennels will be subdivided into those functions:

* Add Kennel Type – This function allows for the creation of a new kennel type.
* Update Kennel Rate – This function updates existing kennel types Rate across the kennel system.
* Add Kennel – This function assigns a kennel id to a kennel type.
* Remove a Kennel – This function changes the status on a kennel to decommissioned for any reason and is no longer available to be rented out.
* Kennel Enquiry – This function check to see if there is any kennels with the selected type available to be rented out on the dates, they wish to enquiry about.

The management of the booking is handled using:

* Make Booking – This function checks to see if a kennel is available on the dates requested and if some gets the customers detail and makes the booking.
* Cancel Booking- This function will find the customers booking at already exists in the system and remove it.
* Record Arrivals- This function checks in the dogs into the kennel system.
* Record Departure- This function checks the dogs out of the kennel system.

System Analysis incorpoRates the following:

* Revenue Analysis- This function adds all the cost of booking for any given year and displays them in a easy to read graph.
* Kennel Analysis- This function gets all the kennel types rent over any given year and displays them in a easy to read graph.

# Functional Components

# User Requirements

## The KennelSYS will perform Kennel administration.

* + 1. KennelSYS will add a new Kennel type.
    2. KennelSYS will update a Kennel type.
    3. KennelSYS will add a kennel.
    4. KennelSYS will remove a Kennel.
    5. KennelSYS will Enquiry a Kennel type.

## KennelSYS will manage Bookings.

* + 1. KennelSYS will make a Booking.
    2. KennelSYS will cancel a Booking.
    3. KennelSYS will list the Arrivals.
    4. KennelSYS will list the Departure

## KennelSYS will perform Administration.

* + 1. KennelSYS will perform a yearly revenue analysis.
    2. KennelSYS will perform a Kennel Type Analysis.

# System Requirements

The System Requirements for KennelSYS are outlined in the system case diagram and user case diagrams below.

# System Level Use Case Diagram

KennelSYS will allow for the management of the boarding Kennels. This including adding, removing and updating kennels. The system will also allow for the process of making and cancelling bookings and record arrivals and departure dates. It also allows for the payment to be taken when the booking has been placed. Analysis will also be incorpoRated within the system.

Manager

Customer

KennelSYS

## 4.1 Kennels

This component includes functions for the adding, updating and removal of kennels in the system. It also includes a function to submit an enquiry about the kennels.

### Add Kennel Type

This function records the details of a kennel type and its Rate on the system. Once a kennel is made on the system it is then available to be used.

Manager

<<includes>>

<<extends>>

|  |  |
| --- | --- |
| Activity Diagram: Add a Kennel Type | |
| Manager | System |
| Confirm Kennel Details  Enter Kennel Details  Invoke Process Kennel Type | Save Kennel Details  Display UI  Display Confirmation Message  No  Yes  Valid  Load Kennel Type UI  Reset UI  Error Message  Validate Kennel Details |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Kennel Type** | |
| **Use Case Id** | 4.1.1 | |
| **Priority** | high | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function defines the kennel type and Rate for a kennel. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the Add Kennel Type.  **Step 4:** The manager enters the Kennel type details:   * Kennel\_type * Description * Rate   **Step 5:** The manager confirms that the kennel Rate is to be set | **Step 2:** The system loads the Kennel type UI.  **Step 3:** Displays UI  **Step 6:** The system validates the data entered:   * All fields must be entered * Kennel Type must be alphabetic characters * Kennel Type must not be already added * Rate must be numeric and greater than zero.   **Step 7:** The system saves the kennel type details in the **Rates File**.   * Kennel\_type * Description * Rate   **Step 8:** The system displays a confirmation message.  **Step 9:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 6:** A blank field detected.  **Step 7:** The system displays an appropriate error message.  “All field must be filled.”  **Step 8:** Return cursor to offending field and return to step 3. |
| **Rate non-numeric or < zero** |  | **Step 6:** A numeric value below Limit was detected.  **Step 7:** The system displays an appropriate error message.  “Value must be greater than zero.”  **Step 7:** The system displays an appropriate error message.  “Rate must be of numeric value”  **Step 8:** Return cursor to offending field and return to step 3. |
| **Type already exist** |  | **Step 6:** Repeated type detected.  **Step 7:** The system displays an appropriate error message.  “Kennel type already exists on the system.”  **Step 8:** Return cursor to offending field and return to step 3. |
| **Conclusions** | The new Kennel Type is added to the **Rates file.** | |
| **Post conditions** | A kennel can now be added to this kennel type | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Update a Kennel Rate

This function updates the details of an existing kennel Rate on the system. Once the update is complete the kennel is made available to be used once again.

Manager

<<includes>>

<<extends>>

|  |  |
| --- | --- |
| Activity Diagram: Update a Kennel Type | |
| Manager | System |
| update Kennel Type details  Confirm Kennel Details  Select Kennel Type  Invoke Process Kennel Type | Display selection Kennel Type details UI  Display Confirmation Message  update Kennel Type details  No  Yes  Valid  Validate Kennel Details  Error Message  Display UI  Retrieves the Kennel Type Details  Reset UI |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Kennel Type** | |
| **Use Case Id** | 4.1.2 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function updates a Rate for a kennel type. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the Update Kennel Type function.  **Step 4:** The manager selects the kennel type to be updated.  **Step 6:** The manager updates the required details:   * Description * Rate   **Step 7:** The manager confirms that the kennel type is to be updated | **Step 2:** The system retrieves all kennel types from the **Rates File**.  **Step 3:** The system Displays UI  **Step 5:** The system displays the details of the selected kennel type for updating.  **Step 8:** The system validates the data entered:   * All fields must be entered * Rate must be numeric and greater than zero.   **Step 9:** The system updates the kennel type details in the **Rates File.**   * Kennel\_type * Description * Rate   **Step 10:** The system displays a confirmation message.  **Step 11:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 7:** A blank field detected.  **Step 8:** The system displays an appropriate error message.  “All field must be filled.”  **Step 9:** Return cursor to offending field and return to step 5. |
| **Numeric value below limit.** |  | **Step 7:** A numeric value below Limit was detected.  **Step 8:** The system displays an appropriate error message.  “Value must be greater than zero.”  **Step 9:** Return cursor to offending field and return to step 5. |
| **String value detected** |  | **Step 7:** String detected in Rate field.  **Step 8:** The system displays an appropriate error message.  “String detected in Rate field! Please enter a numeric value”  **Step 9:** Return cursor to offending field and return to step 5. |
| **Conclusions** | The Kennels Type will be updated on the **Rates file.** | |
| **Post conditions** | The kennel type will have updated Rate and description. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Add kennel

This function allows the manager to add a kennel to the system.

Manager

|  |  |
| --- | --- |
| Activity Diagram: Add Kennel | |
| Manager | System |
| Enter Kennel details  Confirm Kennel Details  Select Kennel Type  Invoke Process Add Kennel | Reset UI  Display Confirmation Message  Save Kennel Details  Set kennel status  Display UI  Retrieves the Kennel Type Details |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Kennel** | |
| **Use Case Id** | 4.1.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function adds a kennel to the system. | |
| **Preconditions** | A Kennel Type must already exist on the system. | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the Add Kennel function.  **Step 5:** The manager select the Kennel Type  **Step 6:** The manager confirms that the kennel type is correct. | **Step 2:** The system retrieves the Kennel Type from the **Rates** **file.**  **Step 3:** The system assigns the next kennel id.  **Step 4:** Displays UI  **Step 7:** The system sets the kennel status as “available”.  **Step 8:** The system saves the kennel detail to the **kennel file**:   * Kennel\_ID * Kennel\_type * Status = ‘A’   **Step 9:** The system displays a confirmation message.  **Step 10:** The system resets the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The kennel has been added to the kennel file. | |
| **Post conditions** | This kennel can now be rented. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Remove a Kennel

This function removes the details of an existing kennel type and its Rate on the system.

Manager

|  |  |
| --- | --- |
| Activity Diagram: Remove Kennel | |
| Manager | System |
| Confirm Kennel Details  Enter Kennel id  Invoke Process Remove Kennel | Display Kennel Details  Retrieves the Kennel Details  Change Status  Update Kennel Status  Display Confirmation Message  Reset UI  Display UI |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Remove Kennel** | |
| **Use Case Id** | 4.1.4 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function removes a kennel from use. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the remove Kennel Type.  **Step 3:** The manager enters the Kennel details.   * Kennel\_ID   **Step 6:** The manager confirms that the kennel is correct. | **Step 2:** Displays UI  **Step 4:** The system retrieves the kennel from the **kennel file.**  **Step 5:** The system displays the kennel details.  **Step 7:** Status changed to “decommissioned”.  **Step 8:** The system updates the **kennel file.**   * Status = ‘D’   **Step 9:** The system displays a confirmation message.  **Step 10:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The kennel status will be put as decommissioned. | |
| **Post conditions** | The kennel will no longer be in uses. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Kennel Enquiry

This function lists all kennels registered on the system regardless of status. The function can filter the list to show only currently available or decommissioned kennels.

Receptionist

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Kennel Enquiry** | |
| **Use Case Id** | 4.1.5 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | None | |
| **Description** | This function allows the Manager to check if a kennel is available to be rented out. | |
| **Preconditions** | A kennel must have been registered on the system before an enquiry can be performed. | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The Manager invokes the Kennel Enquiry function.  **Step 3:** The Manager selects a status to filter the listing:   * All * Available * Decomissioned   . | **Step 2:** System retrieves details of all kennels registered on the system from the Kennel File in order of Kennel\_ID and Displays on UI  Step 4: System retrieves details of all kennels with the selected status from the Kennel File in order of Kennel\_ID and Displays on UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | Specific kennel type availability is known. | |
| **Post conditions** | A kennel type must be registered before its details can be displayed. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Bookings

This component includes functions for the making and cancelling booking.

This component also records arrivals and departure dates.

### Make Booking

This function allows the Manager to make a booking for a kennel.

Customer

Receptionist

|  |  |
| --- | --- |
| Activity Diagram: Make Booking | |
| Manager | System |
| Enter Customer details  Select Customer Type  Confirm booking  Enter info if any  Enter Booking details  Invoke Process Make Booking | Generates next Booking ID  Error Message  No  Yes  Valid  Display Customer details UI  Display Customer UI  Generates next Customer ID  Display Add info UI  Calculate bill  Display Booking UI  Error Message  No  Yes  Valid  Reset UI  Display Confirmation  Sends confirmation email  Save booking details  assign customerID  Save customer details |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Make Booking** | |
| **Use Case Id** | 4.2.1 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Customer | |
| **Description** | This function allows the Manager to make a booking for an available kennel. | |
| **Preconditions** | A kennel must have been registered on the system before a booking can be performed.  There must not be a dog already booked in or for the kennel | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The Manager invokes the make booking function.  **Step 5:** The Manager enters the booking details.   * Kennel\_type * Arrival\_date * Departure\_date   **Step 6:** Manager confirms booking is to proceed  **Step 8:** Manager chooses the kennel number from the list.  **Step 11:** The Manager enters the  Additional details if any.   * CustFirstName * CustLastName * PhoneNo * DogBreed * DogName * Info * CustEmail   **Step 12:** The Manager confirms the booking. | **Step 2:** The system generates the next booking ID.  **Step 3**: System retrieves Kennel Types from the Rates file and loads on the UI.  **Step 4:** The system displays the booking UI.  **Step 7:** The system finds all kennels of the specified type for the dates required from the **Kennel File** and the **Booking File** and loads on UI  **Step 9:** The system calculates the cost of the ( no of days \* Rate) and displays on UI  **Step 13:** System validates the input data:   * All fields must be entered except info field. * Email must be valid format * PhoneNo only numeric values.   **Step 14: System sets booking status to ‘Booked’ (‘B’)**  **Step 13:** The system saves the details in the Booking File:   * BookingID * Kennel\_ID * Arrival\_Date * Departure\_Date * CustFirstName * CustLastName * PhoneNo * CustEmail * DogBreed * DogName * Info * Cost * Status   **Step 21:** The system sends a confirmation message to the customers e-mail address.  **Step 22:** The system displays a confirmation message.  **Step 23:** The system clears the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Numeric value and or special characters detected** |  | **Step 15:** Numeric value and or special characters detected in the First\_Name or Last\_Name field.  **Step 16:** The system displays an appropriate error message.  “Numeric value and or special characters detected. Please re-check first name and or last name.”  **Step 17:** Return cursor to offending field and return to step 14. |
| **Characters detected in mobile field** |  | **Step 15:** Characters detected in mobile field.  **Step 16:** The system displays an appropriate error message.  “Mobile must be numeric value. Please re-enter mobile number.”  **Step 17:** Return cursor to offending field and return to step 14. |
| **Numeric value and or special characters detected** |  | **Step 5:** Numeric value and or special characters detected in the DogType field  **Step 6:** The system displays an appropriate error message.  “Numeric value and or special characters detected. Please re-enter full name.”  **Step 7:** Return cursor to offending field and return to step 4. |
| **Kennel already in use.** |  | **Step 5:** Kennel already in use on date requested.  **Step 6:** The system displays an appropriate error message.  “That kennel is already being used on that date.”  **Step 7:** Return cursor to offending field and return to step 4. |
| **Conclusions** | The dog will be booked in a kennel. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Cancel Booking

This function allows the Manager to cancel a booking for a kennel.

Customer

Receptionist

|  |  |
| --- | --- |
| Activity Diagram: Cancel Booking | |
| Manager | System |
| Confirms the booking details  Choose correct Customer’s Last Name  Enter Customers Last Name  Invoke Process Cancel Booking | Reset UI  Removes booking  Display booking details  System retrieve’s the booking details  Display Customers Details  System retrieve’s customer details  Display UI |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Cancel Booking** | |
| **Use Case Id** | 4.2.2 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Customer | |
| **Description** | This function allows the Manager to cancel a booking. | |
| **Preconditions** | The customer must already booked a kennel. | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The Manager invokes the cancel booking function.  **Step 3:** The Manager searches the customer’s last name.  **Step 6:** The Manager chooses the correct last name.  **Step 9:** The Manager confirms the booking details, | **Step 2:** Display UI.  **Step 4:** The system retrieves customer’s last name from the **Booking file.**  **Step 5:** Displays list of last names.  **Step 7:** The system retrieves the booking details from the **booking file.**   * BookingID * CustFirstName * CustLastName * ArrivalDate * DepartureDate * Email   **Step 8:** The system displays the booking details.  **Step 10:** The system removes the booking from the **booking file.**  **Step 11:** The system displays confirmation that the booking is cancelled.  **Step 12:** The system resets the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The booking will be removed from the **booking file.** | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Record Arrivals

This function checks in the dog in the day of arrival. It updates the kennel file and booking details.

Receptionist

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Record Arrivals** | |
| **Use Case Id** | 4.2.3 | |
| **Priority** | High | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Customer | |
| **Description** | This function lists the arrivals of the customer to the kennel on the day. Also updates the kennel and booking status. | |
| **Preconditions** | The customer must have already booked a kennel. | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The Manager invokes the arrivals function.  **Step 3:** The Manager enters the customer’s last name.  **Step 4:** The Manager confirms the last name to process.  **Step 8:** The Manager chooses the booking for the arrival  **Step 9:** The Manager confirms that is the right person. | **Step 2:** The system Displays the search UI.  **Step 5:** The system searches the **Booking file** for the customer’s last name and the arrival date matching the system date.  **Step 6:** If no bookings found, display message and return to step 3.  **Step 7:** display bookings on UI  **Step 10:** The system updates the **booking file** set status = “Arrived” (A)  **Step 11:** The system updates the **kennel file** set status = “Occupied” (O)  **Step 12:** The system displays a confirmation details.  **Step 13:** The system resets the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The booking status will be updated in the booking file.  The Kennel status will be updated in the kennel file. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Record Departures

This function records the departure of a dog from the kennel.

Also updates the kennel file.

Receptionist

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Record Departure** | |
| **Use Case Id** | 4.2.4 | |
| **Priority** | Medium | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Customer | |
| **Description** | This function records the departure of dog from the kennel on the day. Also updates the kennel status. | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The Manager invokes the departure function.  **Step 3:** The Manager enters the customer’s last name.  **Step 7:** The Manager chooses the booking for the departure  **Step 8:** The Manager confirms that is the right person. | **Step 2:** The system Displays the search UI.  **Step 4:** The system searches the **Booking file** for the customer’s last name and the arrival date matching the system date.  **Step 5:** If no bookings found, display message and return to step 3.  **Step 6:** display bookings on UI  **Step 9:** The system updates the **booking file** set status = “Departed” (D)  **Step 10:** The system updates the **kennel file** set status = “Available” (A)  **Step 11:** The system displays a confirmation details.  **Step 12:** The system resets the UI. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The Kennel status will be updated in the kennel file. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Administration

This function preforms analysis on the revenue to see which month they rent the most kennels.

Also performs a kennel type analysis to see which kennel rents the most over the year.

### Revenue Analysis

This function performs an analysis on the revenue of any given year.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Revenue Analysis** | |
| **Use Case Id** | 4.3.1 | |
| **Priority** | Low | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function displays the revenue analysis of any given year in a graph which the manager come print it out if they wish. | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the revenue analysis function.  **Step 3:** The manager selects the year they wish to analysis.  **Step 4:** The manager confirms the selected year to process.  **Step 6:** The manager selects the print option. | **Step 2:** The system loads the years that have departure dates pasted in the system.  **Step 5:** The System displays the graph.   * Monthly * Sum(cost)   **Step 7:** The system prints out the chart. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The system displays the revenue for the given year.  Print option also available. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Kennel Type Analysis

This function analysis the rent of kennel type over any given year.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Kennel Type Analysis** | |
| **Use Case Id** | 4.3.2 | |
| **Priority** | Low | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This function displays the kennel type analysis of any given year in a graph which the manager come print it out if they wish. | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the Kennel Type Analysis function.  **Step 3:** The manager enters the year they wish to analysis.  **Step 6:** The manager selects the print option. | **Step 2:** The system loads the years that have departure dates pasted in the system.  **Step 4:** The system retrieves details of all bookings for the selected year from the **booking file, kennel file**   * How much a kennel made that year. * Kennel\_type   **Step 5:** System displays the graph in the UI.  **Step 7:** The system prints out the chart. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The system displays and or prints the kennel types for the given year. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

Booking Details

|  |
| --- |
| Customers |

Kennel Details

Kennel System

## Level-1 DFD

P1

Manage Kennel

P2

Manage Booking

Customer

D3

Booking File

Booking Details

Booking Details

Booking Details

Kennel Details

D1

Rates File

Kennel Details

Kennel Details

Kennel Details

D2

Kennel File

Kennel Details

P3

Perform Admin

Kennel Details

Kennel Details

Booking Details

## Level-2 DFD (Process P1: Kennel)

P1.1

Add Kennel Type

D1

Rates File

P1.2

Update Kennel

P1.3

Add Kennel

Kennel Details

Kennel Details

Kennel Details

Kennel Details

Kennel Details

Kennel Details

D2

Kennel File

P1.4

Remove Kennel

Kennel Details

P1.5

Kennel Enquiry

Kennel Details

## Level-2 DFD (Process P2: Booking)

P2.2

Cancel Booking

P2.1

Make Booking

Booking Details

Booking Details

Booking Details

Customer Details

Booking Details

Kennel Details

D2

Kennel File

D3

Booking File

Booking Details

Booking Details

Kennel Details

Customer

Customer

C

P2.4

Record Departure

P2.3

Record Arrivals

Kennel Details

Booking Details

Kennel Details

Booking Details

Booking Details

Booking Details

D2

Kennel File

D3

Booking File

## Level-2 DFD (Process P3: Administration)

Booking Details

D3

Booking File

Booking Details

Kennel Details

D2

Kennel File

P3.2

Kennel Type Analysis

P3.1

Revenue Analysis

# Data Model (Class Diagram)

## Class Diagram

Rates

Kennel\_Type

Description

Rate

1

Can have

🡪

0..\*

Kennels

Kennel\_ID

Status

Bookings

BookingID

Arrival\_Date

Departure\_Date

CustFirstName

CustLastName

PhoneNo

CustEmail

DogBreed

DogName

Info

Cost

Status

1



🡪

0..\*

has a

## Relational Schema

Rates (**Kennel\_Type**, Description, Rate)

Kennels (**Kennel\_ID**, Kennel\_Type, Status)

Bookings (**BookingID**, Kennel\_ID, Arrive\_Date, Departure\_Date, CustFirstName, CustLastName, CustEmail, PhoneNo, DogBreed, DogName, Info, Cost,Status,)

## Database Schema

**Relation:** Rates

Attributes:

Kennel\_Type varchar(4)

Description varchar2 (20) NOT NULL

Rate number(4,2) NOT NULL CHECK > 0

**Primary Key:**  Kennel\_Type

**Relation:** Kennels

Attributes:

Kennel\_Id numeric(2)

Kennel\_Type varchar(4) NOT NULL

Status varchar2(1) NOT NULL

**Primary Key:** Kennel\_Id

**Foreign Key:** Kennel\_Type References Rates

Peter Mc Cafferty – T00095451

**Relation:** Bookings

Attributes:

BookingID numeric (8)

Kennel\_Id numeric (8) NOT NULL

Arrival\_Date date NOT NULL

Departure\_Date date NOT NULL

CustFirstName varchar2 (20) NOT NULL

CustLastName varchar2 (30) NOT NULL

PhoneNo varchar2 (15) NOT NULL

CustEmail varchar2 (50) NOT NULL

DogBreed varchar2 (20) NOT NULL

DogName varchar2 (20) NOT NULL

Info varchar2 (30)

Cost number (5,2) NOT NULL

Status varchar (1) NOT NULL

**Primary Key:**  Booking\_Id

**Foreign Key:** Kennel\_ID References Kennels

# Conclusion

The system KennelSYS as specified in this document includes the functionality needed for the management of kennels and bookings. The system also generates an analysis of yearly income and the most popular kennels in a given year.

Overall the project has progressed well. The functionality included is adequate given the time frame for the assignment. There are additional functions that could be added if the system is to be expanded. For example, more data analysis functions and a customer registration function to consolidate customer bookings.