Kennel System

Submitted By: Aoife Sayers

T-Number: T00170881

Computing With Multimedia – Stage 2

Date Submitted: 25/06/2016

**Table of Contents**

[1. Introduction/overview 3](#_Toc449309936)

[2. Functional Components 4](#_Toc449309937)

[3. Functional User Requirements 5](#_Toc449309938)

[3.1. KennelSYS will perform kennel management. 5](#_Toc449309939)

[3.2. KennelSYS will process bookings. 5](#_Toc449309940)

[3.3. KennelSYS will perform administrative reporting. 5](#_Toc449309941)

[4. Functional System Requirements 6](#_Toc449309942)

[4.1. Manage Kennels 7](#_Toc449309943)

[4.1.1. Set Kennel Type 7](#_Toc449309944)

[4.1.2. Change Kennel Type 9](#_Toc449309945)

[4.1.3. Add a New Kennel 11](#_Toc449309946)

[4.2. Process Bookings 14](#_Toc449309947)

[4.2.1. Make Booking 14](#_Toc449309948)

[4.2.2. Cancel reservation 19](#_Toc449309949)

[4.2.3. Check-in Client 21](#_Toc449309950)

[4.2.4. Check-out Client 23](#_Toc449309951)

[4.3. Process Admin 25](#_Toc449309952)

[4.3.1. List Daily Arrivals 25](#_Toc449309953)

[4.3.2. List Daily Departures 27](#_Toc449309954)

[4.3.3. List Revenue Analysis 28](#_Toc449309955)

[4.3.4. List Kennel Analysis Report 29](#_Toc449309956)

[4.3.5. List Current Kennel Status 30](#_Toc449309957)

[5. System Model 32](#_Toc449309958)

[5.1. DFD Elements 32](#_Toc449309959)

[5.2. Level-0 DFD 33](#_Toc449309960)

[5.3. Level-1 DFD 34](#_Toc449309961)

[5.4. Level-2 DFD (Process P1: Manage Kennels) 35](#_Toc449309962)

[5.5. Level-2 DFD (Process P2: Process Bookings) 36](#_Toc449309963)

[5.6. Level-2 DFD (Process P3: Perform Admin) 37](#_Toc449309964)

[6. Data Model (Class Diagram) 38](#_Toc449309965)

[7. Database Design 39](#_Toc449309966)

[8. Program Specifications 40](#_Toc449309967)

[9. Conclusion 40](#_Toc449309968)

[10. Appendices 41](#_Toc449309969)

[Appendix A 41](#_Toc449309970)

# Introduction/overview

This system is an automated administration system for a doggy day care centre. The system will manage kennel registration and amendments, manage bookings, cancellations and arrival and departures of dogs. The system will also generate admin reports such as daily arrivals/departures, revenue analysis, kennel analysis and a current kennel status.

# Functional Components

The Hierarchy chart representing the functional components of KennelSYS

# Functional User Requirements

The following user requirements have been identified:

## KennelSYS will perform kennel management.

* + 1. KennelSYS will set a kennel type
    2. KennelSYS will change kennel type details
    3. KennelSYS will add a new kennel

## KennelSYS will process bookings.

* + 1. KennelSYS will allow a booking to be made
    2. KennelSYS will allow a booking to be cancelled
    3. KennelSYS will check-in an dog
    4. KennelSYS will check-out a dog

## KennelSYS will perform administrative reporting.

* + 1. KennelSYS will generate a list of daily arrivals
    2. KennelSYS will generate a list of daily departures
    3. KennelSYS will generate a revenue analysis report
    4. KennelSYS will generate a kennel analysis report
    5. KennelSYS will generate a list of the current kennels status

# Functional System Requirements

The main components of KennelSYS are split into 3 components: Manage kennels, process bookings and perform administration.

The diagram below in ***Figure 2*** shows the System Level Use Case Diagram of KennelSYS

KennelSYS

Manager

Customer

***Figure 2***

## Manage Kennels

This component sets kennel type, change kennel type and add a new kennel.

### Set Kennel Type

<< extends >>

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Set Kennel Type** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | | This function defines a kennel type, corresponding rate and description for that kennel type | | |
| **Preconditions** | Manager must fill in a form with the kennel type, a description and a kennel rate. | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager requests the Set Kennel Type function.  **Step 3:** The manager enters the Kennel Type details:   * Type * Description * Rate   **Step 4:** The manager confirms the Kennel Type is to be set. | **Step 2:** The system displays the Set Kennel Type User Interface.  **Step 5:** KennelSYS validates all details entered:   * All fields must be entered * Kennel Type must be a max of 2 alphabetic characters– e.g. S,M,L and should not already exist * Description must not be numeric and less than 30 characters * Rate must be numeric, greater than €0 & less than €99999.99 (5,2)   **Step 6:** The System saves the Kennel Type details in the ***Kennel Type File.***  **Step 7:** The system displays a confirmation message.  **Step 8:** The system clears the User Interface. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Empty Fields/Invalid Data Entered:** |  | **Step 5:** Invalid data is entered.  **Step 6:** The system displays an appropriate error message and focuses on the textbox at fault. |
| **Error Inserting into Database:** |  | **Step 7:** Error inserting into the database.  **Step 8:** An error message is displayed. |
| **Conclusions** | A kennel type is created. | |
| **Post conditions** | Kennels in the Doggy Day Care Centre may now be assigned to this kennel type. | |
| **Business Rules** | Kennel Type must be a unique identifier. | |
| **Implementation Constraints** |  | |

### Change Kennel Type

<< extends >>

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Change Kennel Type** | |
| **Use Case Id** | 2 | |
| **Priority** | 2 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | | This use case changes the details of an existing kennel type. |   Change kennel type allows the primary actor to change the description of and the kennel rate of a kennel type. | |
| **Preconditions** | Manager must fill in a form with the description and kennel rates | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager requests the change Kennel Type function.  **Step 3:** The manager selects the kennel type to be changed.  **Step 5:** The manager modifies the details:   * Description * Rates   **Step 6:** The manager confirms the change of the kennel type. | **Step 2:** The system retrieves the kennel type, rate and description of all kennel types from the ***Kennels Types File*** and displays on the UI in order of rate lowest to highest.  **Step 4:** The system displays the selected types’ description and rate in a text box.  **Step 7:** The system validates the data entered:   * All fields must be entered * Kennel Rate must be greater than €0 & less than €99999.99 * Description must not be greater than 30 characters   **Step 8:** The system updates the kennel type details in the ***Kennels Types File***  **Step 9:** The system displays a confirmation message.  **Step 10:** The system clears the User Interface. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Invalid Data/Empty Fields Entered:** |  | **Step 7:** Invalid data/ Empty fields entered.  **Step 8:** The system displays an error message and focuses on the field at fault. |
| **Error updating new details in the database** |  | **Step 9:** Error updating the new kennel details.  **Step 10:** The system displays an error message |
| **Conclusions** | The kennel type details are changed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Add a New Kennel

Manager

|  |  |  |
| --- | --- | --- |
| Activity: Add a New Kennel |  | |
| **Manager** | **System** | |
| Displays UI  Confirm Details to be added  Select Kennel Type | Assigns Kennel No.  Displays Error Message  Display Confirmation Message  Store Kennel Details in Kennels File  Set Kennel Status=’A’ (available)  N  Retrieves kennel Types from Kennels types File  Data Available  Y |

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add A New Kennel** | |
| **Use Case Id** | 3 | |
| **Priority** | 3 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | This use case adds a new kennel with a unique no. and existing kennel type | |
| **Preconditions** | A kennel type must exist. | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager requests the Add New Kennel function.  **Step 4:** Manager selects a Kennel Type | **Step 2:** The system determines the next Kennel Number from the ***Kennels File***  **Step 3:** The system retrieves the details of all kennel types including the description from the ***Kennels Types File*** and displays on the UI in order of the rate (lowest to highest)  **Step 5:** The system sets the kennel status must to ‘A’ – Available  **Step 6**: The system saves the new kennel details in the ***Kennels File***  **Step 7:** The system displays a confirmation message with the relevant details.  **Step 8:** The system resets the User Interface. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Error retrieving from database:**  **No data available in the database** |  | **Step 2:** The system cannot retrieve the next kennel number from the database.  **Step 3:** The system displays an error message  **Step 3:** No kennel type details available in the ***Kennel Types File***.  **Step 4:** The system displays an error message |
| **Error inserting into database:** |  | **Step 7:** New Kennel not added. The system displays an error message. |
| **Conclusions** | A new kennel has been added. | |
| **Post conditions** |  | |
| **Business Rules** | A Kennel No must be unique. | |
| **Implementation Constraints** |  | |

## Process Bookings

This component of the System makes bookings, cancels bookings, checks in a client or checks out a client.

### Make Booking

<< extends >>

Manager

Client

|  |  |  |
| --- | --- | --- |
| Activity: Make Booking |  | |
| **Manager** | **System** | |
| Selects Dates & Kennel Types  Confirm Types & Date  Validates Details  Valid?  N  Display Booking Details  Confirm Booking  Enters Customer Details  Confirm Customer Details  Y  N  Valid?  Display Confirmation Message | Assigns Kennel No available on dates. Calculates & Displays customer Details UI  Display confirmation message & Resets UI  Insert Booking into ***Bookings File***  Set Kennel Status=’O’  Y  Y  N  Available?  Checks Availability  Displays user interface  Gets Kennel Types from ***Kennels File*** & current date |

Gets next available booking no. from ***Bookings File***

Display Error Message

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Make Booking** | |
| **Use Case Id** | 4 | |
| **Priority** | 4 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Client | |
| **Description** | |  | | --- | |  |   A client makes a booking at the kennels. | |
| **Preconditions** | A manager must enter in the owners details into a form | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager requests the Make reservation function  **Step 4:** The manager selects the some details on the kennel required:   * Kennel Type * Arrival Date * Departure Date   **Step 4:** The manager confirms the requirements entered  **Step 7:** The manager enters the customer details to make the booking:   * Forename * Surname * Street * Town * County * Mobile No   **Step 7:** The manager confirms the make booking details | **Step 2:** KennelSYS assigns the next booking number from the ***Bookings File*** and displays the UI.  **Step 3:** Kennel Type Details are retrieved from the ***Kennels File*** and are displayed in a combo box. Today’s date is also determined and the minimum date of a booking arrival date is set to today and the minimum date of departure is set to 24 hours greater than today.  **Step 5:** The system validates the details entered:   * The system ensures a kennel type is picked * The Date of departure is not before the date of arrival * A Booking is not greater than 30 days in advance   **Step 6:** The system takes the first kennel no available of the selected kennel type the dates specified from the ***Bookings File***. KennelSYS displays the customer UI with KennelNo in a text box. The system also determines the cost of the kennel no from the ***KennelTypes File*** for the no. of days selected & displays it in a text box.  **Step 8:** The system validates the customer’s details entered:   * All Fields must be entered * Forename & Surname must be alphabetical characters and must be less than 20 characters * Street must be of numeric and alphabetical characters and must be less than 30 characters * Town must be alphabetical characters and less than 20 characters * County must be alphabetical characters and less than 20 characters * Mobile No must be less than 10 alpha-numeric characters.   **Step 9:** The Kennel Status for this kennel is set ‘O’ (occupied) for the selected dates. The CheckedIn status in the ***Bookings File*** is set to ‘No’  **Step 10:** The details are saved in the ***Bookings file***  **Step 11:** The system outputs a message to the user with the relevant details  **Step 12:** The UI is cleared. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Unable to retrieve data from the database** |  | **Step 3:** The System cannot find the next Booking No from the ***Kennels File*** and displays an error message***.***  **Step 5:** Kennel Type Details cannot be retrieved from the ***Kennels Types File***. An error message is displayed.  **Step 8:** The System cannot find any kennels available for the selected date. An error message is displayed. |
| **Empty Fields/Invalid Data** |  | **Step 8:** Customer details entered are not valid. An error message is displayed. The program returns and focuses on the field that’s invalid. |
| **Cannot insert into Bookings File** |  | **Step 11:** An error message is displayed describing the issue. |
| **Conclusions** | A booking is made for a client. | |
| **Post conditions** |  | |
| **Business Rules** | Booking no. must be unique  A date of arrival of a booking must be greater than 24 hours in advance.  Date of arrival cannot be after the date of departure.  Cannot make a booking greater than 30 days in advance.  Cannot select a date in the past | |
| **Implementation Constraints** |  | |

### Cancel reservation

<< extends >>

Client

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Cancel Reservation** | |
| **Use Case Id** | 5 | |
| **Priority** | 5 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Client | |
| **Description** | |  | | --- | | A client cancels a reservation at the kennels. | | |
| **Preconditions** | A client must fill out a form confirming that they are cancelling a reservation.  A client must provide the client number or client forename and client surname. | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager requests the cancel reservation function  **Step 3:** The manager selects the name or booking no radio button & searches for specified surname &/forename or booking no.  **Step 7:** The manager selects a reservation to be deleted.  **Step 8:** The manager confirms deletion of reservation | **Step 2:** KennelSYS displays the User Interface  **Step 4:** The system validates the customer’s details entered:   * Surname or Forename or both must be entered or when searching by name * BookingNo must be numeric and greater than 0 * Surname & Forename must be alphabetic characters and less than 20 characters * Booking No. must be entered when searching by booking no. * Booking No. must be numeric, no smaller than 0 and no greater than 9999   **Step 6:** KennelSYS retrieves all customer reservations with this surname&/forename or booking no.  **Step 9:** The system deletes the corresponding reservation and updates the ***Bookings File***  **Step 10:** The UI is cleared. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Cannot retrieve Bookings from *Bookings File*** |  | **Step 7:** Cannot access data from the database. An error message is displayed. |
| **Invalid Details/ Empty Fields entered** |  | **Step 5:** Customer details entered are not valid. An error message is displayed. The program returns and focuses on the field that’s invalid. |
| **Conclusions** | A booking is cancelled for a client. | |
| **Post conditions** |  | |
| **Business Rules** | More than 24 hours cancellation notice from the date of arrival  A client must provide a surname/forename or a booking no to cancel a booking | |
| **Implementation Constraints** |  | |

### Check-in Client

Client

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Check-in Client** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** | Client | |
| **Description** | |  | | --- | | This use case checks-in a client. | | |
| **Preconditions** | The manager must enter the client’s forename and surname or client number.  A Check In must have a booking made. | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** A client arrives at doggy day care  **Step 2:** The manager invokes the check-in client function.  **Step 5:** The manager can search by surname for a specific person or can select a booking to check to in  **Step 7:** The manager confirms the client check in. | **Step 3** KennelSYS displays the UI  **Step 4:** KennelSYS displays all the bookings where the date of arrival matches the current date  **Step 6:** KennelSYS retrieves all reservations made with this surname from the ***Bookings File*** where the date of arrival matches today’s date.  **Step 8:** KennelSYS updates the booking field Checked In from ‘No’ to ‘Yes’ in the ***Bookings File.***  **Step 9:** The UI is reset. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Error accessing the database** |  | **Step 8:** Error the ***Booking File*** cannot be updated. |
| **Conclusions** | The manger checks-in a client and receives a payment | |
| **Post conditions** |  | |
| **Business Rules** | A booking must exist to be checked in.  A booking can only be checked in on the date of arrival specified – not beforehand or after. | |
| **Implementation Constraints** |  | |

### Check-out Client

Client

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Check-out client** | |
| **Use Case Id** | 1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | | This use case checks-out a client. | | |
| **Preconditions** | A booking must have a status of CheckedIn = ‘yes’ to check out a client and the date of departure must be the current date. | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** A client arrives at doggy day care  **Step 3:** The manager selects a reservation.  **Step 4:** The manager confirms the details and confirms the client check out | **Step 2:** KennelSYS displays the UI and displays all the bookings where checked in = ‘yes’ and date of departure = today’s date.  **Step 5:** The booking is deleted from the ***Bookings File***and the room no. gets a status=’A’ in the ***Kennels File***  **Step 6:** KennelSYS outputs a confirmation dialog confirming the client has been checked out  **Step 7:** The UI is cleared. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  | **Step 6:** Error the ***Bookings File*** cannot be deleted. KennelSYS outputs a message. |
| **Conclusions** | A client is checked out. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Process Admin

This component will list daily arrivals, list daily departures, list the revenue analysis, list a kennel analysis and lists the current kennel status.

### List Daily Arrivals

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Daily Arrivals** | |
| **Use Case Id** | 8 | |
| **Priority** | 8 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | | The manager requires a list of all clients arriving on that day. | | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the List Daily Arrivals Function  **Step 2:** The manager selects a date from the date time picker and confirms the selection | **Step 3:** The System retrieves details of the date of arrivals on the selected date. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Failure to access database** |  | **Step 4:** The system fails to retrieve the details of all bookings on the selected date from the ***Bookings File***  The system displays an error message. |
| **Conclusions** | A listing of all daily arrivals on a selected date at KennelSYS is generated. | |
| **Post conditions** |  | |
| **Business Rules** | Client listing is displayed and cannot be amended or deleted. | |
| **Implementation Constraints** |  | |

### List Daily Departures

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Daily Departures** | |
| **Use Case Id** | 9 | |
| **Priority** | 9 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | |  |   The manager requires a list of all clients departing on that day. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the List Daily Arrivals Function  **Step 2:** The manager selects a date from the date time picker and confirms the selection | **Step 3:** The System retrieves details off date of arrivals on the selected date. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Failure to access database** |  | **Step 3:** The system fails to retrieve the details of all clients from the ***Bookings File.***  **Step 4:** The system displays an error message. |
| **Conclusions** | A listing of all daily arrivals at KennelSYS is generated. | |
| **Post conditions** |  | |
| **Business Rules** | Client listing is displayed and cannot be amended or deleted. | |
| **Implementation Constraints** |  | |

### List Revenue Analysis

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Revenue Analysis** | |
| **Use Case Id** | 10 | |
| **Priority** | 10 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | |  | | The manager requires a list of the revenue analysis. | | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the List Revenue Analysis function.  **Step 2:** The manager selects a year from the combo box. | **Step 3:** The system calculates the revenue generated for each month of the year from the ***Bookings File.*** |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Error Retrieving Data from database** |  | **Step 4:** The system fails to retrieve the details of all clients from the ***Admin File.***  **Step 5:** The system displays an error message. |
| **Conclusions** | A listing of all revenue for a specified year for KennelSYS is generated. | |
| **Post conditions** |  | |
| **Business Rules** | Revenue listing is displayed and cannot be amended or deleted. | |
| **Implementation Constraints** |  | |

### List Kennel Analysis Report

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Kennel Analysis Report** | |
| **Use Case Id** | 11 | |
| **Priority** | 11 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | | The manager requires a list of the kennels analysis. | |  | | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the List Kennel Analysis function. | **Step 2:** The system retrieves each the kennel types from the ***Kennel Types File,*** calculates the no. times booked and the sum of the revenue generated from the ***Bookings File*** for the current year. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Error Retrieving Data from database** |  | **Step 3:** The system fails to retrieve the details from the ***Kennels File***& the ***Bookings File***.  **Step 4:** The system displays an error message. |
| **Conclusions** | A listing of all the kennel analysis at KennelSYS is generated | |
| **Post conditions** |  | |
| **Business Rules** | Kennel analysis listing is displayed and cannot be amended or deleted. | |
| **Implementation Constraints** |  | |

### List Current Kennel Status

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **List Current Kennel Status** | |
| **Use Case Id** | 12 | |
| **Priority** | 12 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | |  | | --- | | The manager requires a list of the current occupied and vacant kennels. | |  | | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Typical Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The manager invokes the List Kennel Analysis function. | **Step 2:** The system retrieves all kennels from the ***Kennels File*** where the kennel status = ‘o’ and adds it to the occupied kennels list.  **Step 3:** The system retrieves all kennels from the ***Kennels File*** where the kennel status = ‘v’ and adds it to the Vacant Kennels list. |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Error Retrieving Data from database** |  | **Step 4:** The system fails to retrieve the occupied/vacant kennel details from the ***Kennels File***& the ***Bookings File***.  **Step 5:** The system displays an error message. |
| **Conclusions** | A listing of all the current kennel status at KennelSYS is generated | |
| **Post conditions** |  | |
| **Business Rules** | Kennel analysis listing is displayed and cannot be amended or deleted. | |
| **Implementation Constraints** |  | |

# System Model

## DFD Elements

**External Entities:**

Client

**Data Stores:**

D1 Kennels Types File

D2 Kennels File

D3 Bookings File

**Processes:**

**P1 Manage Kennels**

P1.1 Set Kennel Type

P1.2 Change Kennel Type

P1.3 Add a new Kennel

**P2 Process Bookings**

P2.1 Make reservation

P2.2 Cancel reservation

P2.3 Check-in Client

P2.4 Check-out client

**P3 Perform Admin**

P3.1 List Daily Arrivals

P3.2 List Daily Departures

P3.3 List Revenue Analysis

P3.4 List Kennel Analysis Report

P3.5 List Current Kennel Status

The following dataflow diagrams have been produced for the system:

## Level-0 DFD

Kennel Details

Receipt

Client

Client Details

Kennel

SYS

## Level-1 DFD

Kennel Type

Details

P1

Manage Kennels

Kennel Type

Details

D1

Kennels Type File

Kennel Type

Detail

Kennel No

Kennel Details

P3

Process Admin

Kennel Details

D2

Kennels File

Booking

Details

Kennel Details

P2

Process Bookings

Kennel Type

Details

D1

Kennels Type File

Booking Details

Booking Details

D3

Bookings File

## Level-2 DFD (Process P1: Manage Kennels)

Sets Kennel Type

Change Kennel Type

Client

P1.2

Change Kennel Type Details

Updated Kennel Type

Kennel Type Details

P1.1

Set Kennel Type

D1

Kennels Type File

Kennel Types

Kennel

Type

Details

Add New Kennel

Client

P1.3

Add a New Kennel

D2

Kennels File

New Kennel

Details

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

D2

Kennels File

Kennels No

D1

Kennels Types File

Kennels Type Details

P2.3

Check-In

Client

Booking No

P2.1

Make Booking

Check–in details

New Booking Details

Client’s Booking Details

Booking Details

Checked In client details

Client

Client

D3

Bookings File

Booking

Details

Checked-In Client’s Details

Check–out details

Cancel Booking

P2.2

Cancel Reservation

P2.4

Check-Out

Client

Deletes Booking

Details

Client’s Check-Out Details

## Level-2 DFD (Process P3: Perform Admin)

P3.2

List Daily Departures

Booking details

P3.1

List Daily Arrivals

Booking details

P3.4

List Kennel Analysis

Kennel

Analysis Details

Revenue

Analysis

Details

D3

Bookings File

Kennels Types details

D1

Kennels Types File

P3.3

List Revenue Analysis

P3.5

List Current Kennel Status

Kennels details

Kennels details

D23

Kennels File

# Data Model (Class Diagram)

has

Booking

BookingNo{pk}

Surname

Forename

DateFrom

DateTo

Street

Town

County

MobileNo

CheckedIn

Cost

KennelType

KennelType{pk}

Description

Rate

1

has

1

1..\*

Kennel

KennelNo{pk}

Type

KennelStatus

1..\*

# Database Design

# Relational Schema

KennelType (**KennelType**, Rate, Description)

Kennel (**KennelNo**, Type, KennelStatus)

Booking(BookingNo, KennelNo, Surname ,Forename , DateFrom, DateTo, Street, Town, County, MobileNo, CheckedIn, Cost

* 1. **Database Schema**

Relation KennelTypes

KennelType char(2) NOT NULL

Rate numeric(5,2) NOT NULL

Description char(30) NOT NULL

Primary Key: KennelType

Relation Kennels

KennelNo numeric(4) NOT NULL

Types char(2)

KennelStatus char(2) NOT NULL

Primary Key: KennelNo

Foreign Key: Types References KennelTypes

Relation Bookings

BookingNo numeric(4) NOT NULL

KennelNo numeric(4)

Surname char(20)

Forename char(20)

DateFrom date

DateTo date

Street varchar(30)

Town char(20)

County char(20)

MobileNo varchar(10)

CheckedIn char(3)

Cost numeric(5,2)

Primary Key: BookingNo

Foreign Key: KennelNo References Kennels

# Program Specifications

List Daily Arrivals

List Daily Departures

List Revenue Analysis

List Current Kennel Status

# Conclusion

In conclusion, KennelSYS has implemented all the modules as presented on the functional components hierarchy chart as seen in the [Functional System Requirements](#_Functional_System_Requirements) part 3. The system as a whole works, however the system could have better exception handling. Also the listing for the Revenue Analysis was not implemented as this document had specified. The system only displayed data for the months on which there were bookings and left out the remainder of the months.

# Appendices

## Appendix A