



# GYMSYS

**Software Engineering**

**Submitted By: Alan Devlin (TL\_KCOMP\_GY6)**

**Submitted on: 12 December 2013**

## Table of Contents

<b>1. Introduction</b>	1
<b>2. User Requirements</b>	2
2.1. The system will provide membership type management:	3
2.2. The system will provide member management:	3
2.3. The system will provide purchases administrations:	3
2.4. The system will provide gym manager administrations:	3
<b>3. System Requirements</b>	4
3.1 Create Membership Type	4
3.2 Delete Membership Type	6
3.3 Change Membership Type	8
3.4 Print Membership Type	10
3.5 Create New Member	12
3.6 Change Member Profile	14
3.7 Delete Member	16
3.8 View Member Profile	18
3.9 Purchase Membership	20
3.10 List Expiring Memberships	22
3.11 View Membership Lists	24
3.12 View Revenue Analysis	26
<b>4. System Model</b>	28
Level 0 DFD:	29
Level 1 DFD:	30
Level 2 DFD:	31
P1 Membership Types Processes	31
P2 Members Processes	32
P3 Purchases Processes	33
P4 Transaction Processes	34
<b>5. Data Model</b>	35

5.1 Relational Model .....	36
<b>6. Database Schema .....</b>	<b>37</b>
<b>7. Program Specifications .....</b>	<b>39</b>
7.1 Create Membership Type .....	39
7.2 Create New Member.....	43
7.3 Edit Member Details .....	50
<b>8. Conclusion.....</b>	<b>56</b>

# 1. Introduction

GymSYS is a membership registration system which will allow a gym to be able to register a member, allocate them to the required membership type, and process their payment.

GymSYS allows the user to create new, edit existing, and delete old membership types. It is important for the system to have this capability to allow flexibility in the gym's membership types. It allows the user to create a new membership type, for example a 3-month special, with the hope of boosting sales. GymSYS also serves the capability of being able to edit a membership type. This is particularly important if the gym wants to reduce/increase the cost of the membership type. If a membership type is no longer in use, GymSYS provides the functionality of being able to set its status to 'inactive', whereby no members can be allocated to it.

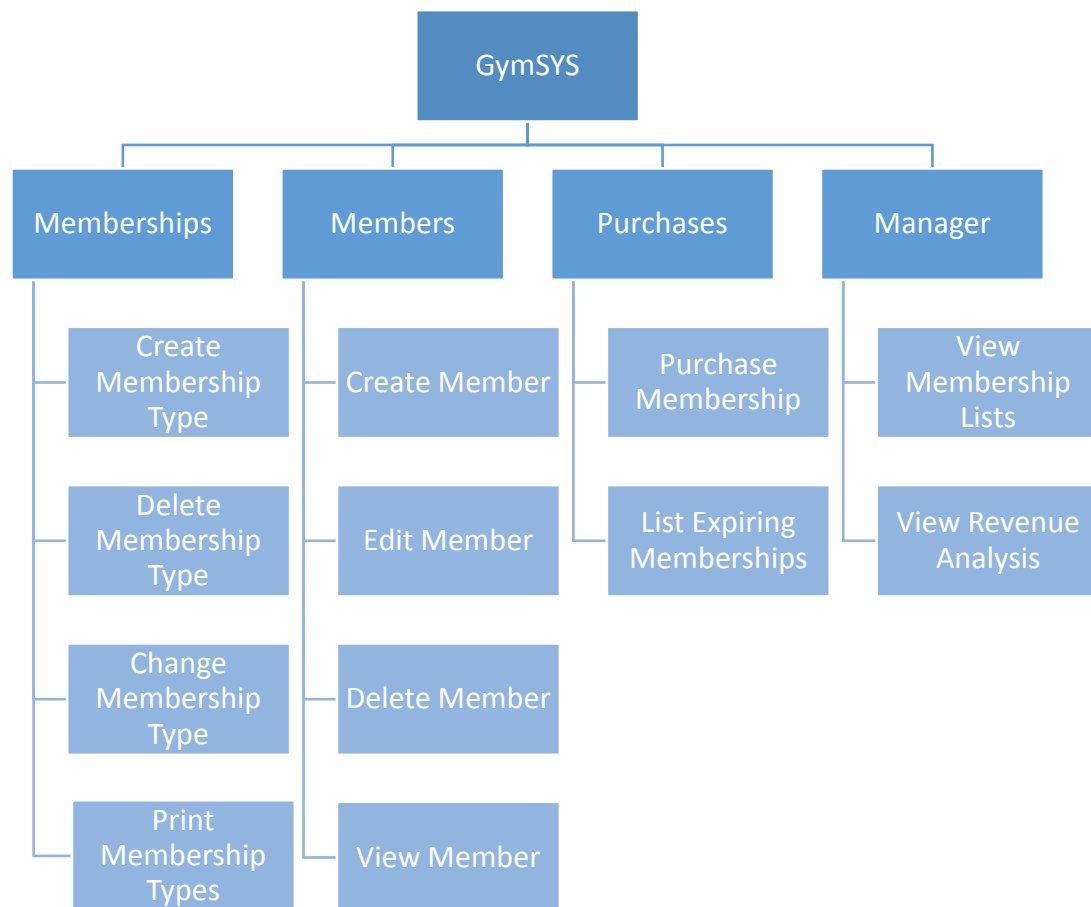
GymSYS allows for registration and altering of its members. It is possible to create a new member, which will be provided with a unique Membership ID. The member is then allocated to the membership type which they have chosen. It will also be possible to register payment details when creating a new member. The GymSYS also provides an Edit Member function which allows the user to edit some of the member's details (e.g. membership type, phone number, address, etc.). If a member decides to depart the gym, GymSYS provides the Delete Member functionality.

GymSYS also allows for purchases management. When a member pay's for a new membership type, the purchase details are entered into the system, which can also allocate when the members membership is expected to expire. This allows for the functionality of allowing the gym manager to be able to view all the members whose memberships are expiring in the upcoming month. The gym is then able to remind these members that their membership is expiring and ask them if they would like to renew.

The final functionality that GymSYS provides is the ability to produce reports. These reports include revenue analysis, which will give an insight into how much money the gym has made in the calendar year. It also provides a list of all members who are currently registered to the gym.

## 2. User Requirements

The functional requirements for the proposed system are shown on the following hierarchy chart:



The user requirements for the system (GymSYS) are described below:

**2.1. The system will provide membership type management:**

- 2.1.1. GymSYS will allow the user to create a new membership type
- 2.1.2. GymSYS will allow the user to delete a membership type
- 2.1.3. GymSYS will allow the user to change a membership type
- 2.1.4. GymSYS will allow the user to print membership types

**2.2. The system will provide member management:**

- 2.2.1. GymSYS will allow the user to create a new member
- 2.2.2. GymSYS will allow the user to delete a member
- 2.2.3. GymSYS will allow the user to edit a member
- 2.2.4. GymSYS will allow the user to view a member

**2.3. The system will provide purchases administrations:**

- 2.3.1. GymSYS will allow the user to create a membership purchase
- 2.3.2. GymSYS will allow the user to list expiring memberships

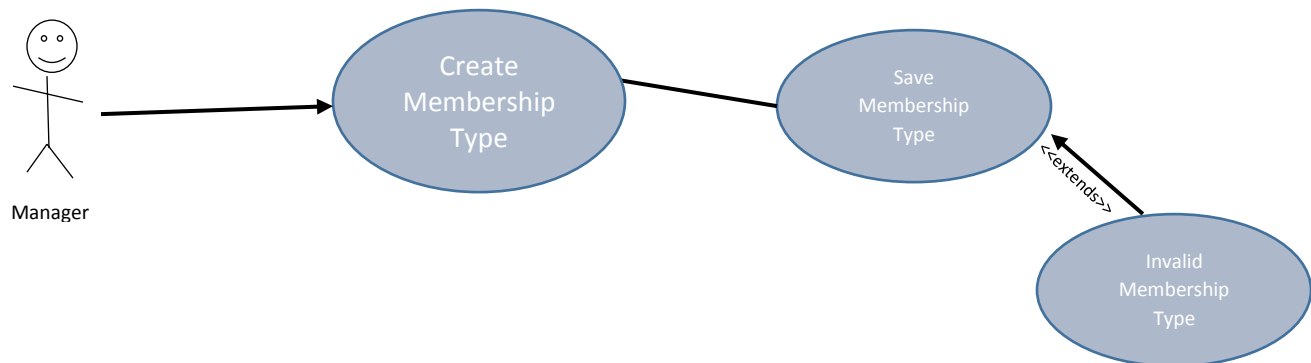
**2.4. The system will provide gym manager administrations:**

- 2.4.1. GymSYS will allow the user to view membership lists
- 2.4.2. GymSYS will allow the user to generate revenue analysis

### 3. System Requirements

#### 3.1 Create Membership Type

##### Use Case: Create Membership Type



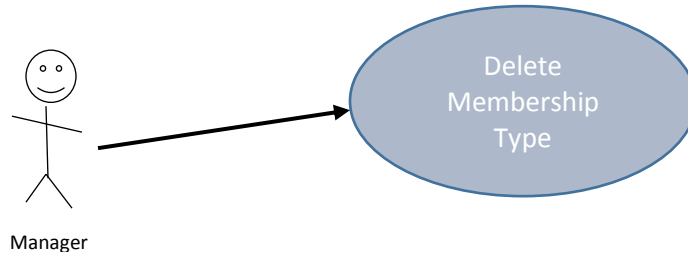
Use Case Name	Create Membership Type		Author: A.D.
Use Case Id			Date: 15/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors			
Description	The manager creates a new type of membership which can then be allocated to new members		
Preconditions	Details of a membership type must be defined before creating membership types. , e.g. duration of membership, description, i.e. student, over-65, and cost		
Trigger	Scenario begins when the manager wants to create a new membership type		
Typical Scenario	Actor Action	System Response	
	<b>Step 1: A form which describes the new membership type must be filled in prior to registration</b>		
	<b>Step 2: The user begins registering the new membership type</b>		
	<b>Step 3: MemType_ID is determined by the user</b>		
	<b>Step 4: The user enters the relevant data:</b>		

	<ul style="list-style-type: none"> <li>• <b>MemType</b></li> <li>• <b>Description</b></li> <li>• <b>Duration</b></li> <li>• <b>Cost</b></li> </ul>	<p><b>Step 5: The system validates the data entered:</b></p> <ul style="list-style-type: none"> <li>• Ensures all data entered is valid</li> <li>• Ensures MemType_ID is unique</li> <li>• Ensures name must not be numeric values</li> </ul> <p><b>Step 6: Default values are assigned:</b></p> <ul style="list-style-type: none"> <li>• Number of registered members is set to zero</li> <li>• MemType status is set to 'active'</li> </ul> <p><b>Step 7: Membership type details are stored in the <i>Membership Type File</i>.</b></p> <p><b>Step 8: Displays a confirmation message on the user interface.</b></p>
<b>Alternate Scenarios</b>	<b>Step 5:</b> Data validation fails –Display Error Message	
<b>Conclusions</b>	A new record is created in the Membership Type File.	
<b>Post conditions</b>	MemType_ID is created New members can be registered under a Membership Type	



### 3.2 Delete Membership Type

#### Use Case: Delete Membership Type

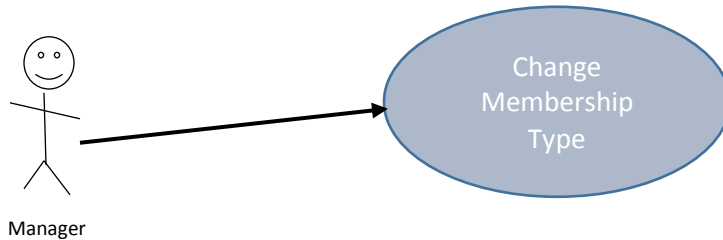


Use Case Name	Delete Membership Type		Author: A.D.
Use Case Id			Date: 15/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors			
Description	The manager can delete a membership type when it becomes obsolete. The gym may have special offers to entice the public in to joining, but these types may be for a limited time only.		
Preconditions			
Trigger	Scenario begins when the user wants to delete a membership type, mainly, due to it having become obsolete.		
Typical Scenario	Actor Action	System Response	
	Step 2: The user must select which MemType they want to delete  Step 4: User must confirm membership type deletion	Step 1: The system retrieves all MemTypes and displays them to the user  Step 3: The systems shows a Warning Message to ensure that the user wants to delete this membership type  Step 5: Membership Type status is set to “Deleted”	
Alternate Scenarios	Step 4: User doesn’t confirm deletion – System returns to Step 2		

<b>Conclusions</b>	The status of the Membership Type is set to “Deleted” in the Membership Type File
<b>Business Rules</b>	Membership types to be deleted can have existing members, but no new members must be assigned to the deleted Membership Type

### 3.3 Change Membership Type

#### Use Case: Change Membership Type

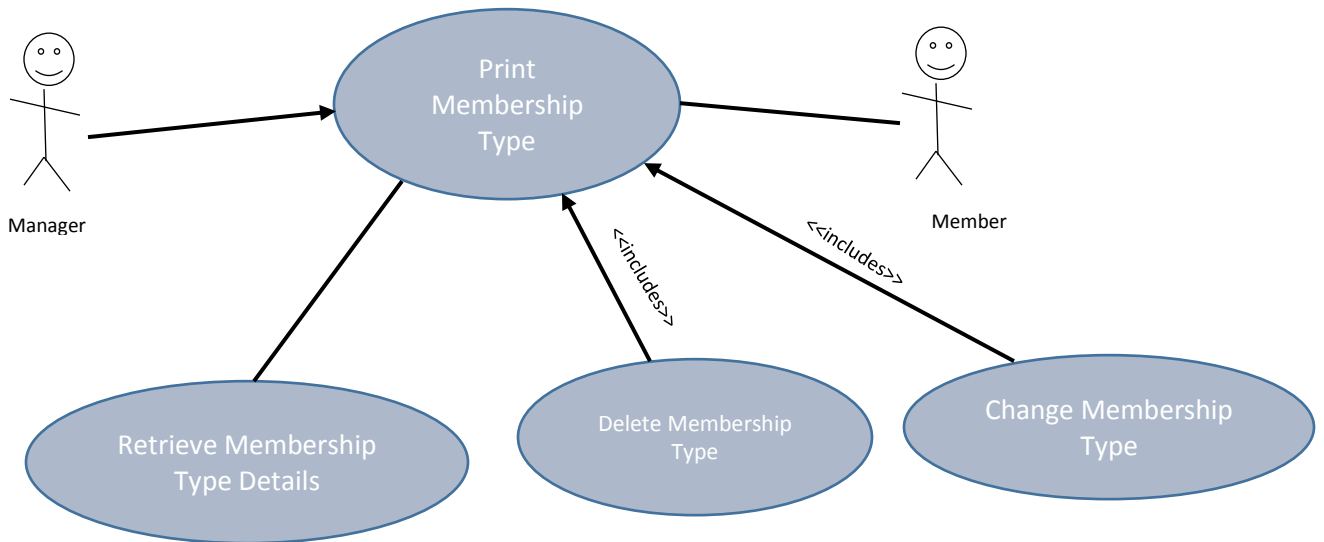


<b>Use Case Name</b>	<b>Change Membership Type</b>		<b>Author: A.D.</b>
<b>Use Case Id</b>			<b>Date: 15/10/13</b>
<b>Priority</b>	1		
<b>Source</b>	Manager		
<b>Primary Business Actor</b>	Manager		
<b>Other Participating Actors</b>			
<b>Description</b>	The manager can make changes to the membership type, allowing for variations in its details, e.g. price may have fallen/risen		
<b>Preconditions</b>	Membership type must have a unique MemType_ID There must be no existing members associated with this Membership Type		
<b>Trigger</b>	Scenario begins when the user wants to edit a membership type.		
<b>Typical Scenario</b>	<b>Actor Action</b>	<b>System Response</b>	
	<b>Step 2: The user selects the membership type they want to edit</b>	<b>Step 1: The system retrieves a summary of all MemTypes currently registered on the system and displays them on the UI</b>	
	<b>Step 4: The user makes the relevant change to the membership type</b> <ul style="list-style-type: none"><li>• Description</li><li>• Duration</li><li>• Cost</li></ul>	<b>Step 3: The system retrieves the current details of the chosen membership type and displays them on the UI</b>	
		<b>Step 5: The system validates the data entered:</b>	

		<ul style="list-style-type: none"> <li>• Ensures all data entered is valid</li> <li>• Ensures description field must not be numeric values</li> <li>• Ensures there are no existing members associated with this Membership Type</li> </ul> <p>Step 6: The system displays a warning message asking the user if they want to proceed with changes</p> <p>Step 7: User confirms changes</p> <p>Step 8: Changes are updated in the <i>Membership Type File</i>.</p>
<b>Alternate Scenarios</b>	<p><b>Step 1:</b> No Membership Types are found – Display Error Message</p> <p><b>Step 4:</b> Invalid Data is entered – Display Error Message</p> <p><b>Step 4:</b> Existing Members found – Changes cannot be made to this Membership Type</p>	
<b>Conclusions</b>	The record belonging to the selected membership type is updated with the relevant changes	
<b>Post conditions</b>	New members can be assigned to the updated membership type	
<b>Business Rules</b>	Duration may only be changed only if there are no existing members of this membership type	

### 3.4 Print Membership Type

#### Use Case: Print Membership Types



Use Case Name	Print Membership Type		Author: A.D.
Use Case Id			Date: 15/10/13
Priority	1		
Source	Member		
Primary Business Actor	Manager		
Other Participating Actors	Potential Member		
Description	The manager can print a list of all membership types. The system is required to produce a list which states, duration of membership and membership cost.		
Preconditions			
Trigger	Potential new member requests membership type list		
Typical Scenario	Actor Action	System Response	
	<b>Step 1: Member requests list of all membership types</b>  <b>Step 2: The manager selects "Print Membership Types"</b>  <b>Step 4: The user can specify if they want the list in a certain order:</b> <ul style="list-style-type: none"><li>• Duration</li><li>• Cost</li></ul>	<b>Step 3: The system retrieves all "active" membership types and displays them on the UI</b>	

	Step 5: The user may requests a print out of the list of membership types	Step 6: This generates a print file which includes all the required data and sends to a print device
Alternate Scenarios	Step 2: The system cannot retrieve membership types – Display an error message	
Conclusions	A print file containing a list of membership types is created	
Business Rules	Only active membership types are viewed in this function	

# GymSYS

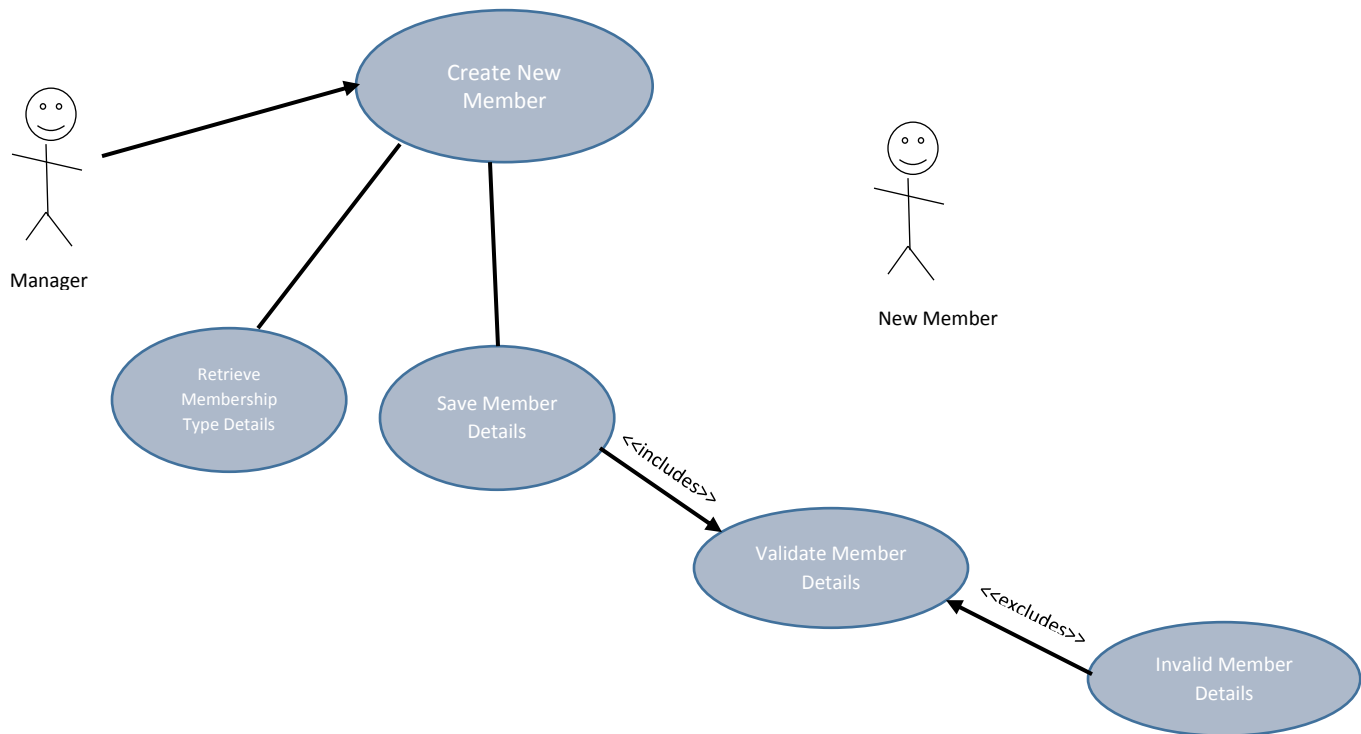
## Membership Rates

MemType_ID	Description	Duration (months)	Cost (€)
M12	Standard	12	600
M6	Standard	6	400
M3	Standard	3	185
M1	Trial	1	65
S12	Student	12	450
S6	Student	6	240
C12	Couple	12	960
C6	Couple	6	580
O12	Over-65	12	400
O6	Over-65	6	230

Image 3.4.1 Example of Membership Type List Printed from GymSYS

### 3.5 Create New Member

#### Use Case: Create New Member



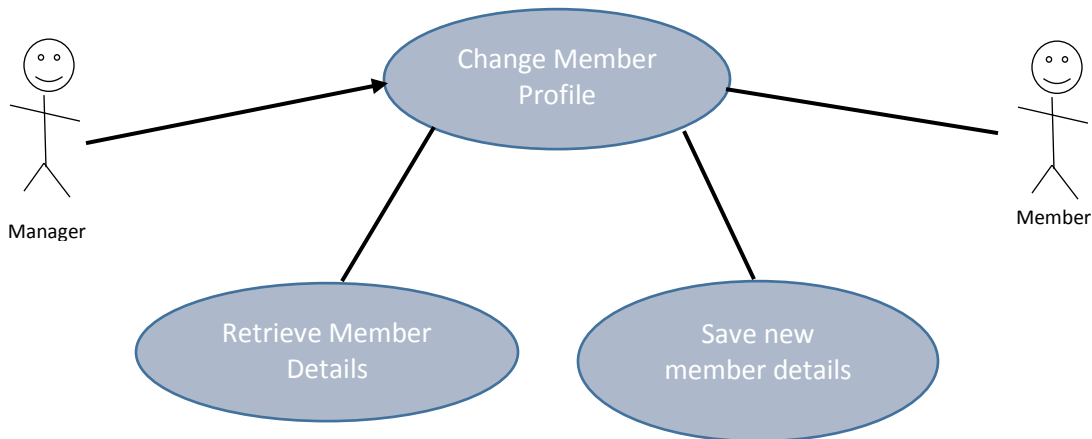
Use Case Name	Create New Member		Author: A.D.
Use Case Id			Date: 15/10/13
Priority	1		
Source	Member		
Primary Business Actor	Manager		
Other Participating Actors	Member		
Description	A new member is created and all their details are saved		
Preconditions	A new member must chose a membership type to avail of and correctly enter all details.		
Trigger	Scenario begins when the member wants to create a membership with GymSYS		
Typical Scenario	Actor Action	System Response	

	<p><b>Step 1: The new member fills out a registration form and choses the membership type that they wish to avail of</b></p> <p><b>Step 4: The user enters the members details including:</b></p> <ul style="list-style-type: none"> <li>• Name</li> <li>• Date of Birth</li> <li>• Address</li> <li>• Membership Type</li> <li>• Purchase Details</li> </ul>	<p><b>Step 2: The system assigns a unique Mem_ID and displays it on the UI</b></p> <p><b>Step 3: The system retrieves the information from the <i>Membership Types File</i> and displays the information on the UI</b></p> <p><b>Step 5: The system calculates the Membership expiry date by using the formula: System Data + Duration of Membership Type</b></p> <p><b>Step 6: The system validates the data entered:</b></p> <ul style="list-style-type: none"> <li>• Ensures all data entered is valid</li> <li>• Generates a unique Mem_ID number for new member.</li> </ul> <p><b>Step 7: Member details are stored in the <i>Member File</i> and purchase details are stored in the <i>Purchases File</i>.</b></p> <p><b>Step 8: Displays confirmation message on the user interface</b></p>
<b>Alternate Scenarios</b>	<b>Step 3: Data validation fails – Invalid Data Entered – Display Error Message</b>	
<b>Conclusions</b>	A new record is created in the Members File and Purchases File	
<b>Post conditions</b>	New members are assigned a unique Mem_ID	



### 3.6 Change Member Profile

#### Use Case: Change Member Profile

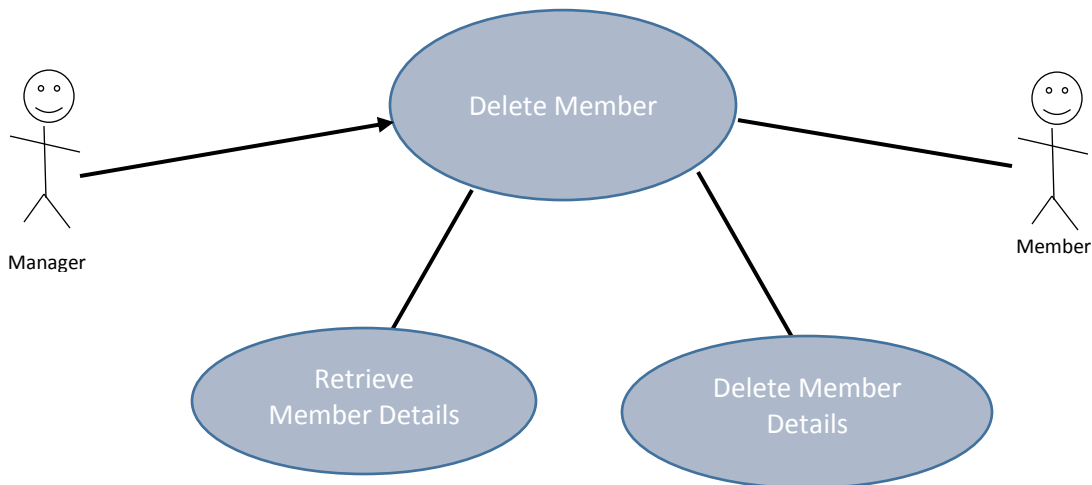


Use Case Name	Change Member Profile		Author: A.D.
Use Case Id			Date: 15/10/13
Priority	1		
Source	Member		
Primary Business Actor	Manager		
Other Participating Actors	Member		
Description	The manager can make changes to the member's profile. This allows the system to make changes to the members details, e.g. Address, Payment Details		
Preconditions	Members must have an existing Mem_ID		
Trigger	Scenario begins when the manager wants to edit a member's profile.		
Typical Scenario	Actor Action	System Response	
	Step 2: The user selects the Member which the wish to make changes to	Step 1: The system lists all Members and displays them on the UI	
	Step 4: The user makes the	Step 3: The system retrieves the current details of the chosen member and displays them on the UI	

	<p>relevant change to the member's profile</p> <p>Step 6: The user confirms changes</p> <p>Step 8: User confirms that they wish to proceed with changes</p>	<p>Step 5: The system validates the data entered:</p> <ul style="list-style-type: none"> <li>• Ensures all data entered is valid</li> <li>• Ensures that the members profile is not currently active and their membership has expired</li> </ul> <p>Step 7: The system displays a warning message asking the user if they want to proceed with changes</p> <p>Step 9: Changes are updated to the member's profile. A new MemType_ID is associated with the member if they have changed membership</p>
Alternate Scenarios	Step 1: System cannot retrieve Members information – Error Message is displayed	
Conclusions	The record belonging to the selected member is updated with the relevant changes	
Post conditions	The MemType_ID which has been allocated to the member may be updated on the members profile if they have changed their membership type	

### 3.7 Delete Member

#### Use Case: Delete Member

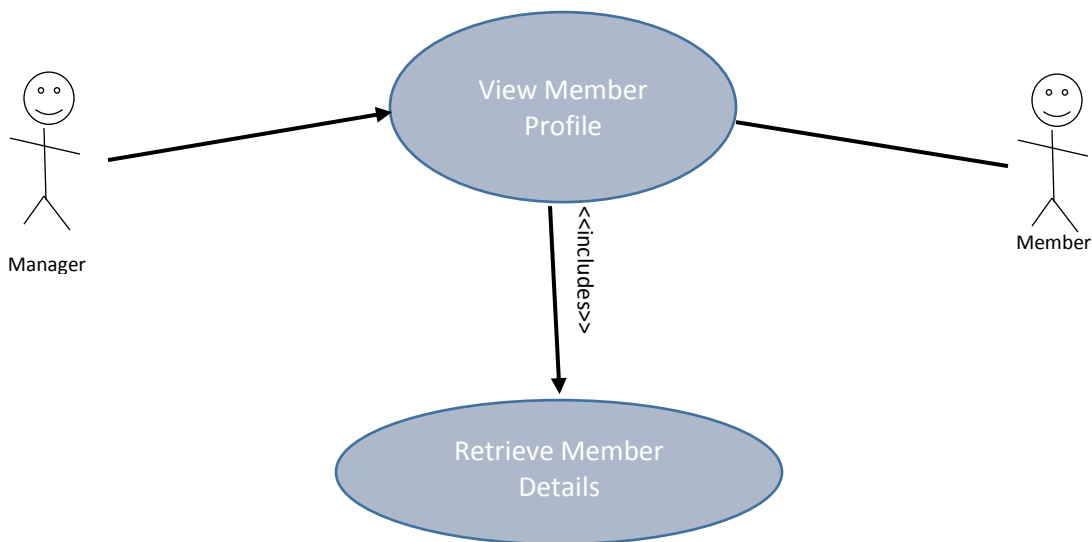


Use Case Name	Delete Member		Author: A.D.
Use Case Id			Date: 29/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors	Member		
Description	The manager deletes the member's profile		
Preconditions	The members profile must not be active		
Trigger	Scenario begins when the member wishes to cancel their membership		
Typical Scenario	Actor Action	System Response	
	<b>Step 1: The member must give written instructions of their wish to cancel their membership</b>	<b>Step 3: The system retrieves the members profile and displays it on the UI</b>	
	<b>Step 2: The manager enters the Mem_ID into the system to retrieve the members profile details</b>		
	<b>Step 4: The manager must select</b>		

	<p>the “Delete Member” button</p> <p>Step 6: User confirms they want to delete profile.</p>	<p>Step 5: The system sends a “Warning Message” to ensure that the user wants to delete the member</p> <p>Step 7: Members profile status is set to “Deleted”</p>
Alternate Scenarios	Step 6: User doesn’t confirm deletion, returns to step 2	
Conclusions	The record corresponding to the Mem_ID is set to “deleted” in the <i>Members File</i> .	

### 3.8 View Member Profile

#### Use Case: View Member Profile

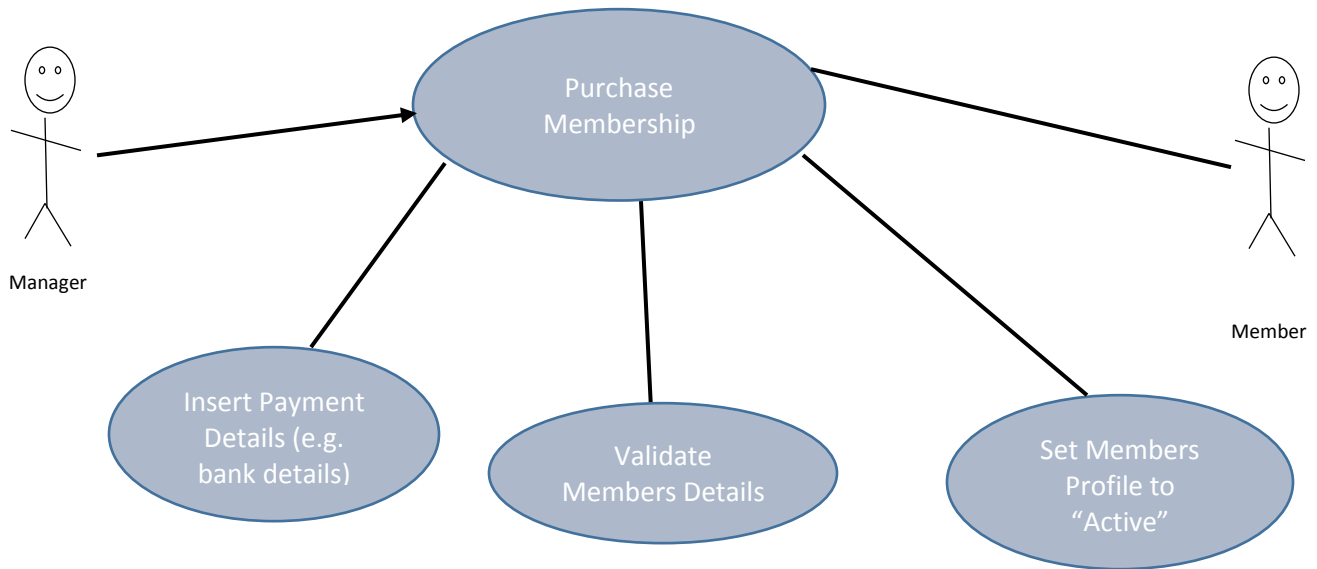


Use Case Name	View Member Profile		Author: A.D.
Use Case Id			Date: 29/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors	Member		
Description	The manager can view the member's profile. Therefore they can see what membership type they are assigned to , their details, etc.		
Preconditions			
Trigger	Scenario begins when the user wants to view the members profile		
Typical Scenario	Actor Action	System Response	
	Step 2: The user enters the relevant Mem_ID	Step 1: The system asks the user to enter the Mem_ID  Step 3: System retrieves the members details from the <i>Members File</i> and relays them back to the user	

<b>Alternate Scenarios</b>	<b>Step 2: The user enters the wrong Mem_ID, return to step 1</b>	
<b>Conclusions</b>	The record corresponding to the Mem_ID type is relayed to the user	
<b>Business Rules</b>	The member must have a valid Mem_ID otherwise the system cannot relay the details back to the user	

### 3.9 Purchase Membership

#### Use Case: Purchase Membership



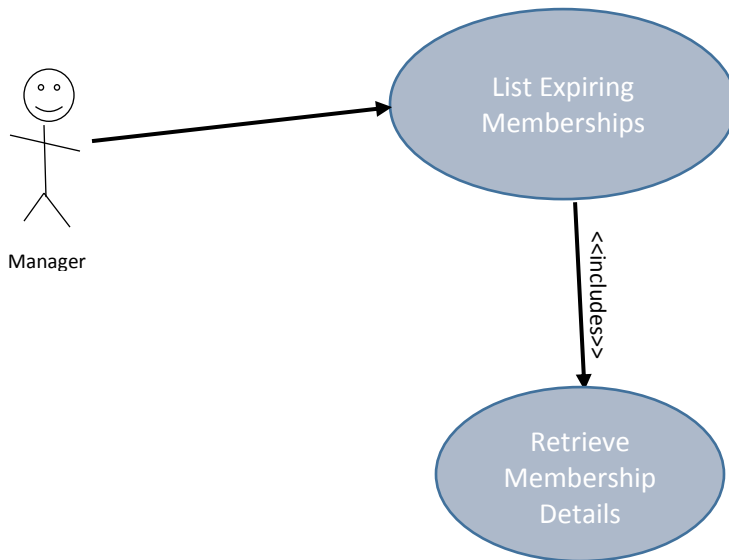
Use Case Name	Purchase Membership		Author: A.D.
Use Case Id			Date: 29/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors	Member		
Description	The member wishes to purchases a membership		
Preconditions	The member must already have been given a Mem_ID and assigned to a relevant MemType_ID The members current membership must be expired		
Trigger	Scenario begins when the member wishes to avail of a membership at GymSYS		
Typical Scenario	Actor Action	System Response	
	Step 1: The member must provide bank details which can be entered into the system  Step 2: The user enters the Mem_ID into the system to generate their details	Step 3: System relays the members details to the user on the UI	

	<p><b>Step 4: The user enters the following in the “Purchase Membership” section:</b></p> <ul style="list-style-type: none"> <li>• <b>Mem_ID</b></li> <li>• <b>Date of purchase</b></li> <li>• <b>MemType_ID</b></li> <li>• <b>Purchase Method</b></li> <li>• <b>Purchase Amount</b></li> <li>• <b>Date of membership expiry</b></li> </ul> <p><b>Step 6: When all data has been validated, the user confirms the payments details</b></p>	<p><b>Step 5: The system validates the data entered:</b></p> <ul style="list-style-type: none"> <li>• <b>Ensures all data entered is valid</b></li> <li>• <b>Ensures the Mem_ID entered is valid</b></li> <li>• <b>Ensures there is a relevant MemType_ID that the member is assigned to</b></li> </ul> <p><b>Step 7: The payment details are then stored in the <i>Purchases File</i>.</b></p>
<b>Alternate Scenarios</b>	<b>Step 2: Mem_ID is invalid, enter again</b>	
<b>Conclusions</b>	The payment details for the member corresponding to the Mem_ID are updated and stored in the <i>Purchases file</i> .	
<b>Post conditions</b>		
<b>Business Rules</b>	The member must have a valid Mem_ID	
<b>Implementation Constraints</b>	The members must not have an “Active” membership, all memberships must have expired	



### 3.10 List Expiring Memberships

#### Use Case: List Expiring Memberships

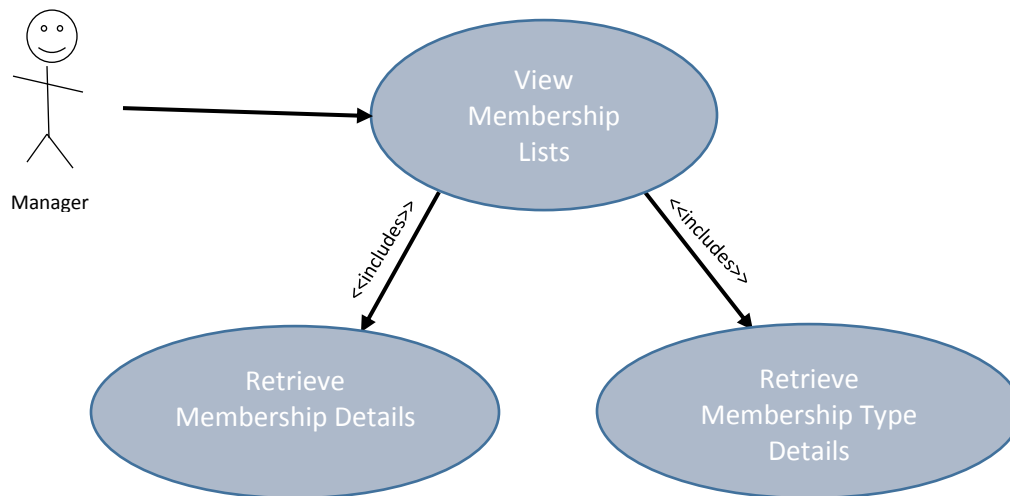


Use Case Name	List Expiring Memberships		Author: A.D.
Use Case Id			Date: 29/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors			
Description	The manager wishes to view all memberships which are expiring in the following month		
Preconditions	All memberships must have been provided with an expiration date which has been allocated when the member is purchasing the membership		
Trigger	Scenario begins when the manager wishes to view all memberships which are due to expire in the next month		
Typical Scenario	Actor Action	System Response	
	Step 1: The manager enters GymSYS and clicks on "List Expiring Memberships"	Step 2: The system retrieves all memberships which are due to expire in the next month	
		Step 3: The system gives the user the option to print the list if	

	Step 4: The manager can choose to print the list or exit the menu	needed
Alternate Scenarios	Step 2: No memberships are due for expiration in the next month, the list is empty	
Conclusions	The system provides a list of all members which have memberships expiring in the next months. The manager is then able to contact theses members to remind them about their expiring memberships and ask them if they would like to extend their memberships	

### 3.11 View Membership Lists

#### Use Case: View Membership Lists

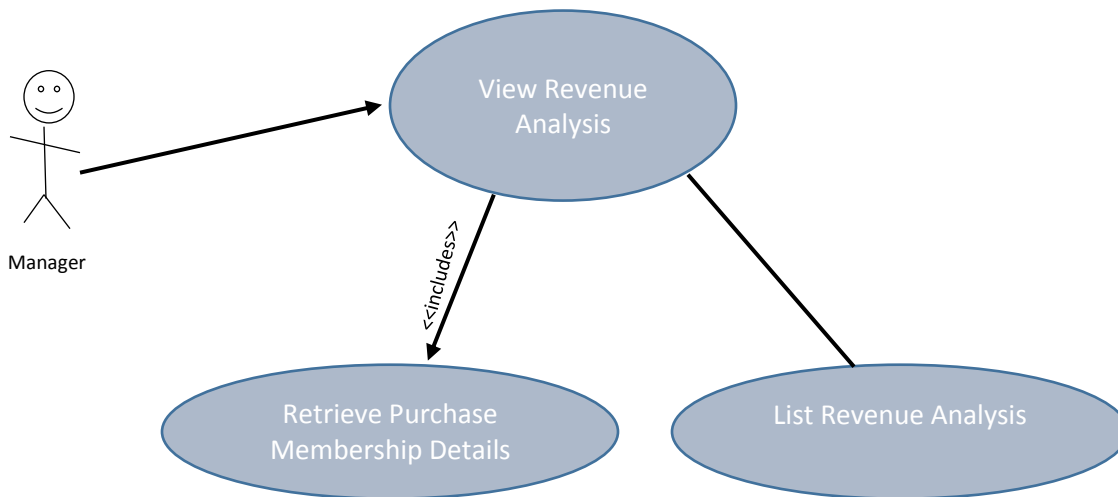


Use Case Name	View Membership Lists		Author: A.D.
Use Case Id			Date: 29/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors			
Description	The manager wishes to view all memberships		
Preconditions	All members must have been provided with a Mem_ID and a MemType_ID		
Trigger	Scenario begins when the manager wishes to view all memberships which are currently active		
Typical Scenario	Actor Action	System Response	
	Step 1: The manager enters GymSYS and clicks on "View Memberships"	Step 2: The system retrieves all memberships which are currently "Active"	
		Step 3: The system gives the user the option to print the list if needed	

	<b>Step 4: The manager can choose to print the list or exit the menu</b>	
<b>Alternate Scenarios</b>	Step 2: The system cannot return any membership details – Error Message Displayed – Return to Step 1	
<b>Conclusions</b>	The system provides a list of all currently “Active” memberships. This allows the manager to see how many members are currently using each membership type. The manager can decide if one membership type is not being used as much as the others, he can do a special offer to entice more customers	

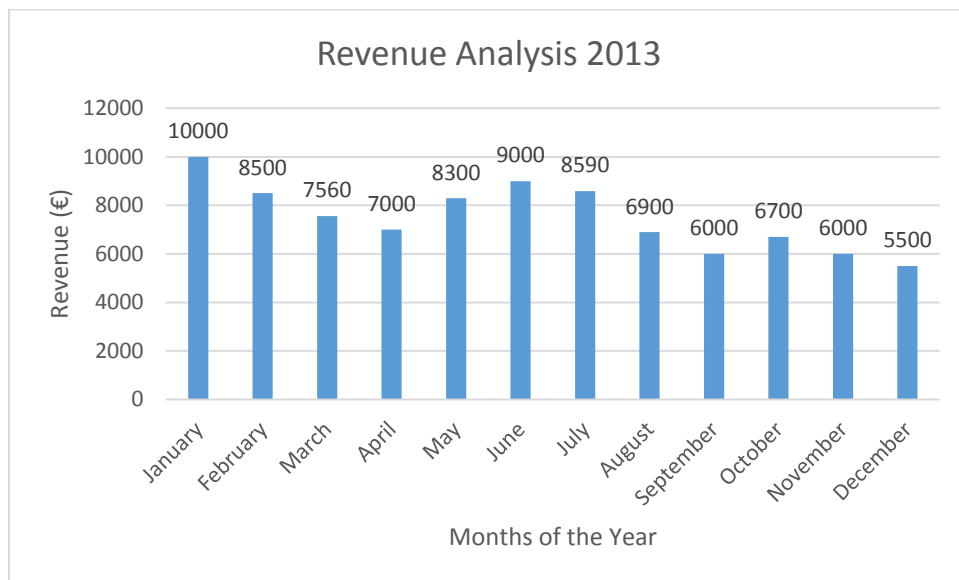
### 3.12 View Revenue Analysis

#### Use Case: View Revenue Analysis



Use Case Name	View Membership Lists		Author: A.D.
Use Case Id			Date: 29/10/13
Priority	1		
Source	Manager		
Primary Business Actor	Manager		
Other Participating Actors			
Description	The manager wishes to view the revenue analysis for the company		
Preconditions			
Trigger	Scenario begins when the manager wishes to view the revenue analysis		
Typical Scenario	Actor Action	System Response	
	Step 1: The manager enters GymSYS and clicks on "View Revenue Analysis"	Step 2: The system retrieves details of all totalled payments which have been made for each month  Step 3: The system presents the option of representing the details in graph form	
	Step 4: The manager can whether		

	he wants the information in a bar graph or a list	
<b>Conclusions</b>	The system provides details of all memberships which have been purchased for each month of the year. This information can be viewed as a list or in a bar graph	
<b>Business Rules</b>	It is only possible to view the details for the current calendar year (January-December)	



**Image 3.12.1 Example of Bar graph which is generated by GymSYS which represents revenue analysis for the year 2013**

## 4. System Model

DFD Elements:

### External Entities

Members

### Data Stores

**D1** Members File

**D2** Membership Type File

**D3** Purchases File

### Processes

**P1** Manage Membership Types

P1.1 Create New Membership Type

P1.2 Delete Membership Type

P1.3 Change Membership Type

P1.4 Print Membership Types

**P2** Manage Members

P2.1 Create New Member

P2.2 Edit Member

P2.3 Delete Member

P2.4 View Member

**P3** Manage Purchases

P3.1 Purchases Membership

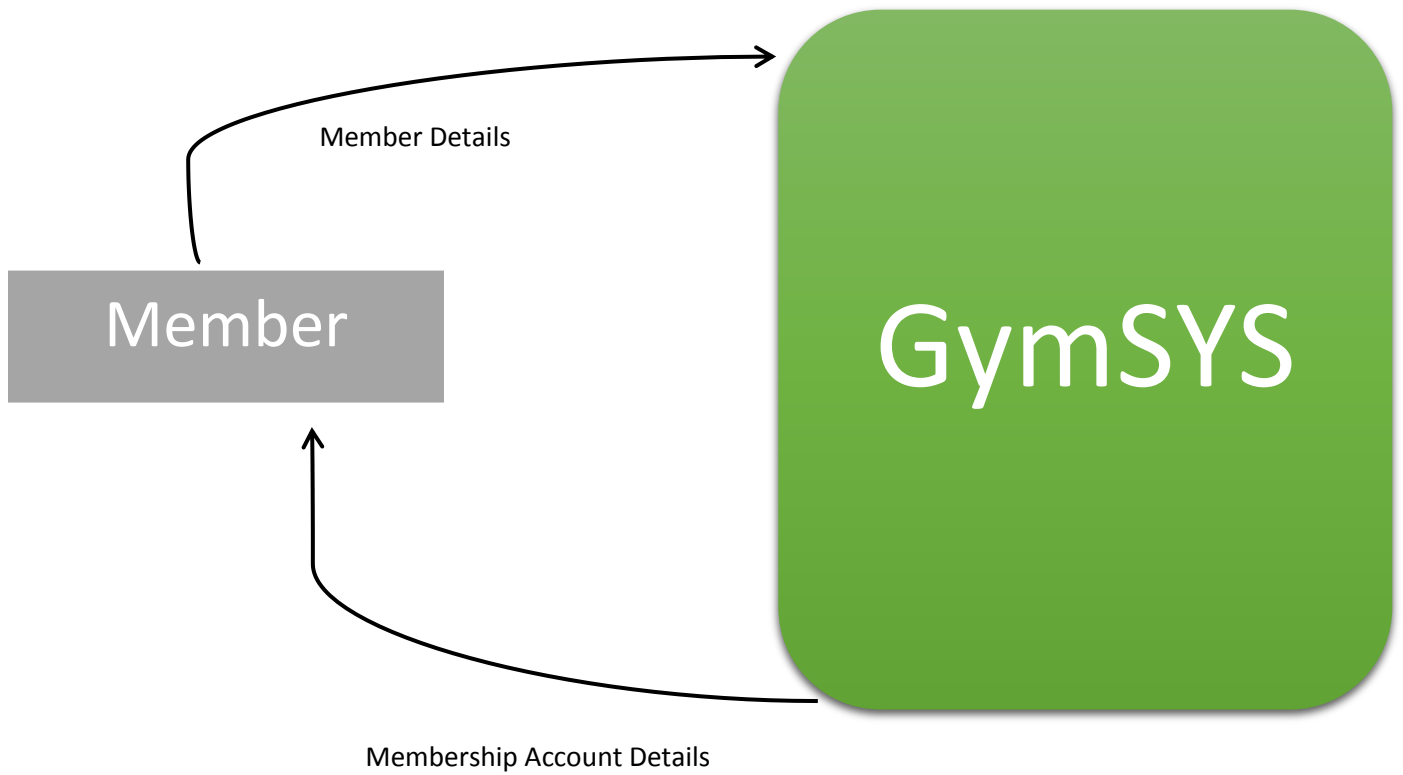
P3.2 List Expiring Memberships

**P4** Reports for Management

P4.1 View Membership Lists

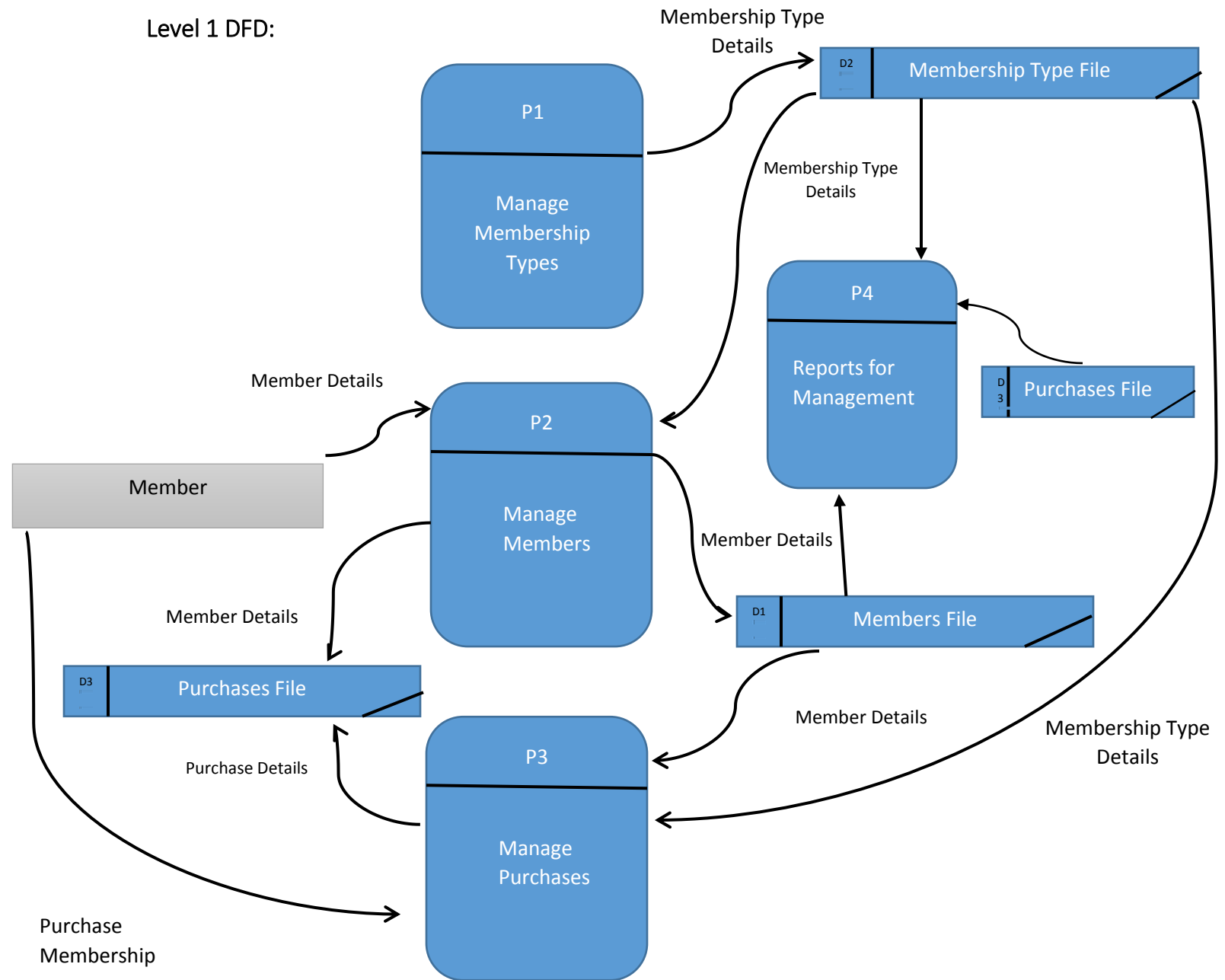
P4.2 View Revenue Analysis

Level 0 DFD:



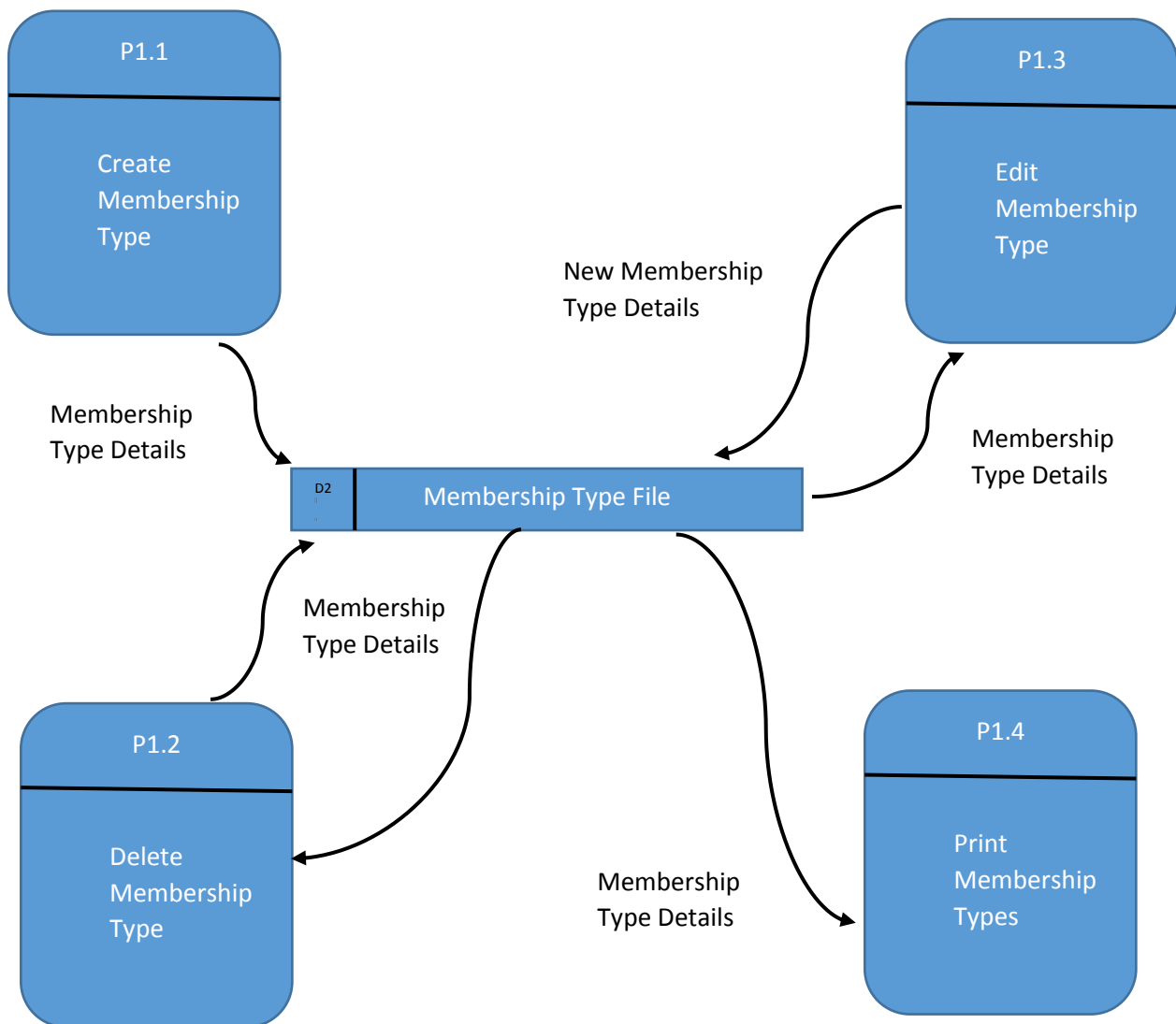


Level 1 DFD:



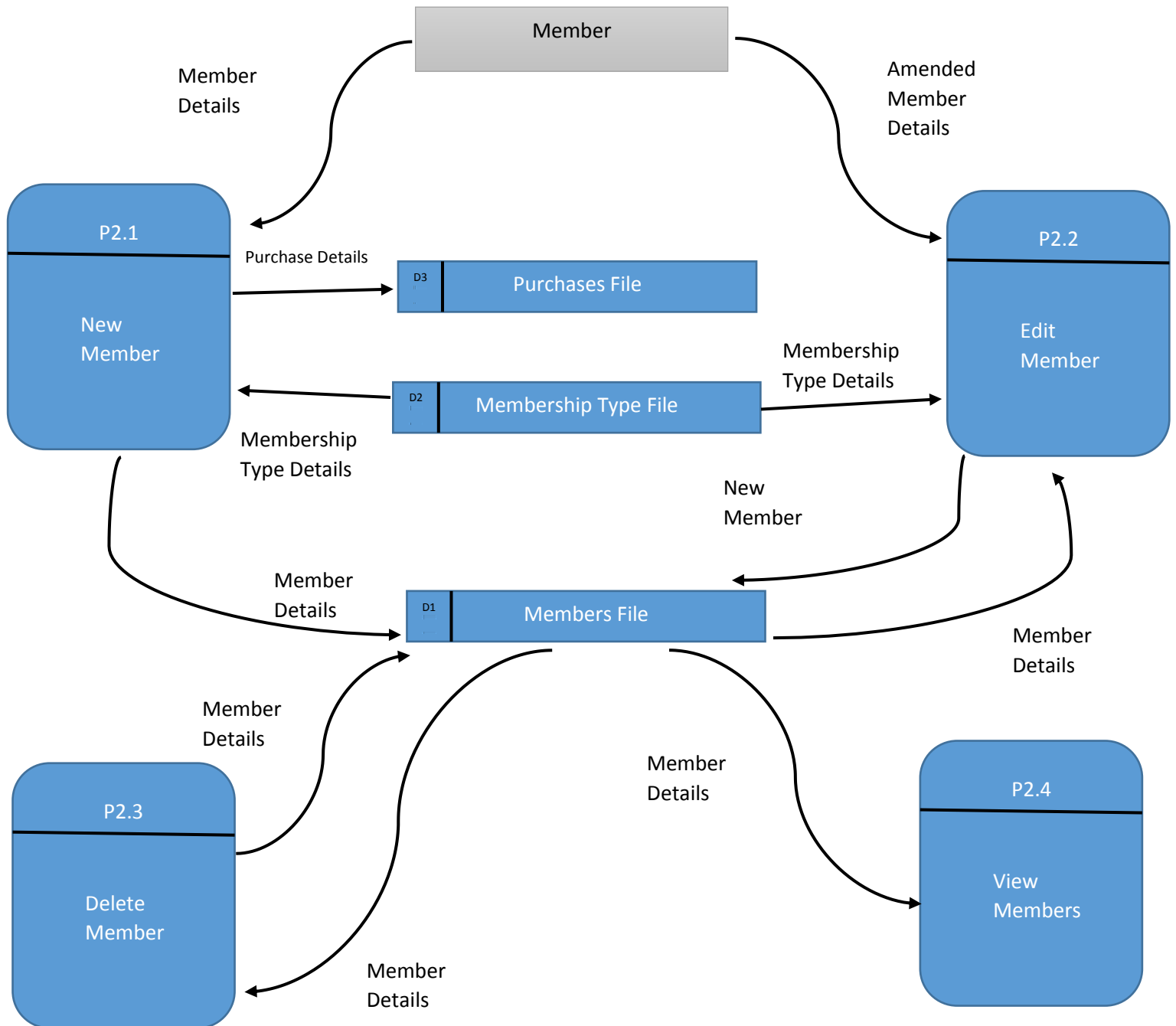
## Level 2 DFD:

### P1 Membership Types Processes

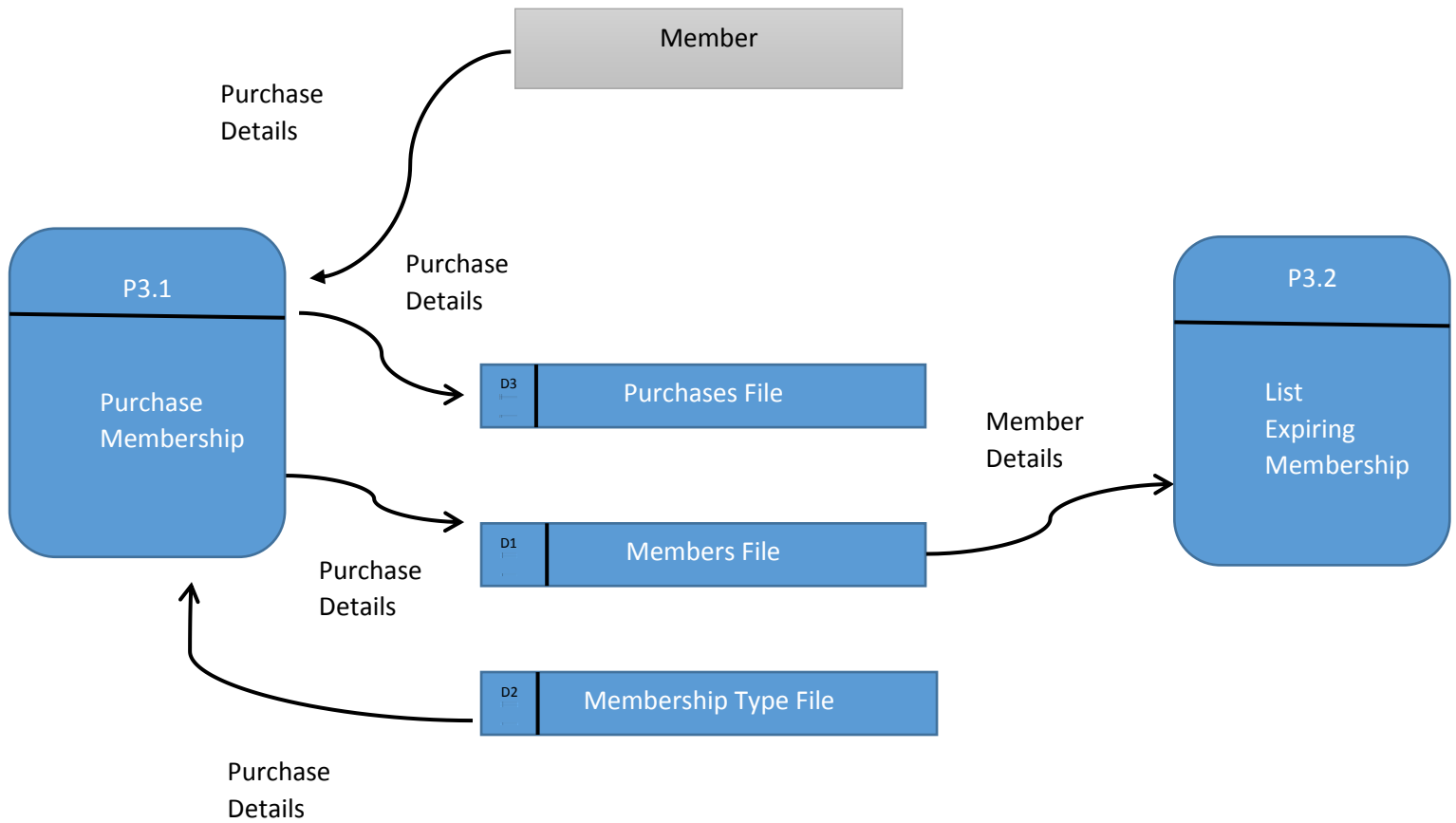


## Level 2 DFD:

P2 Members Processes

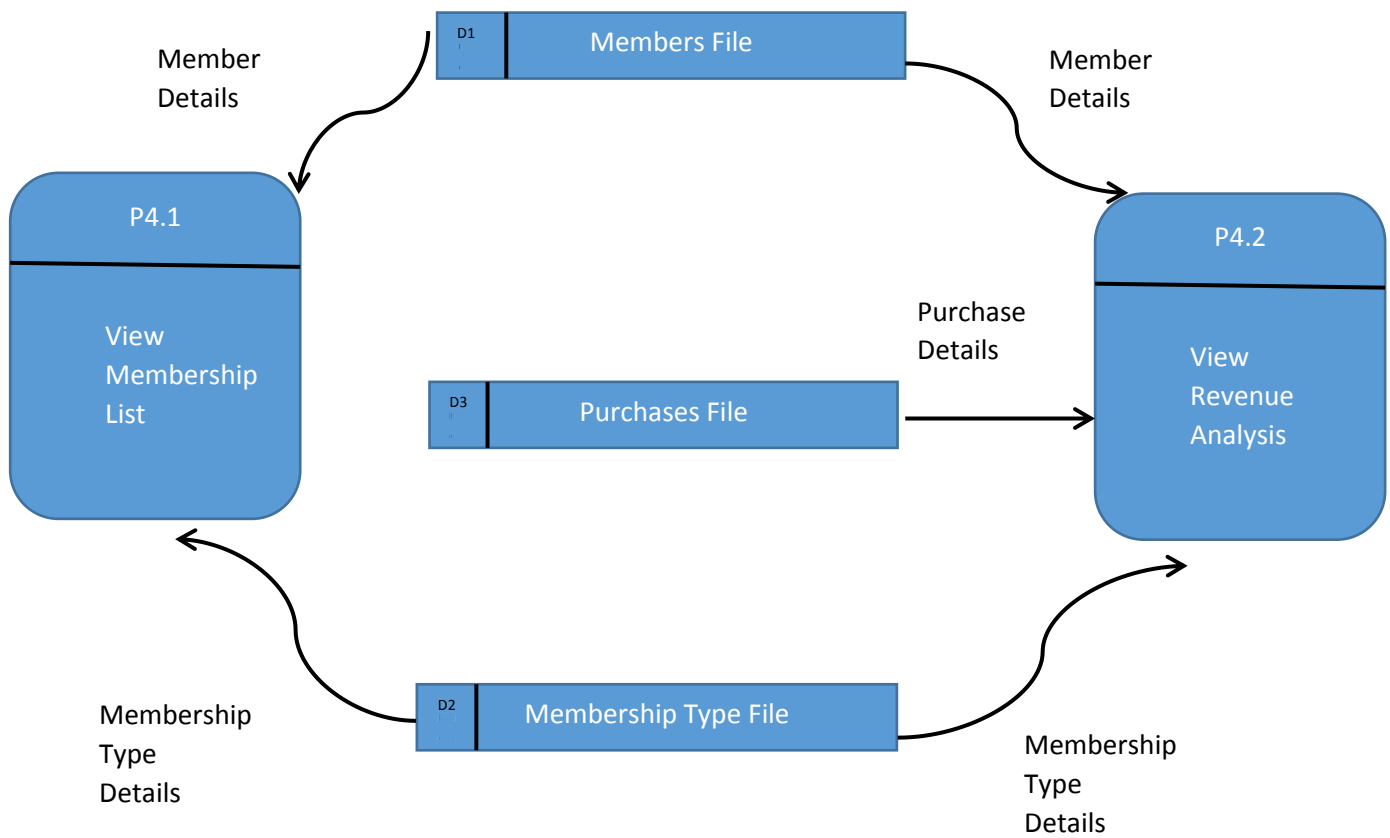


### P3 Purchases Processes

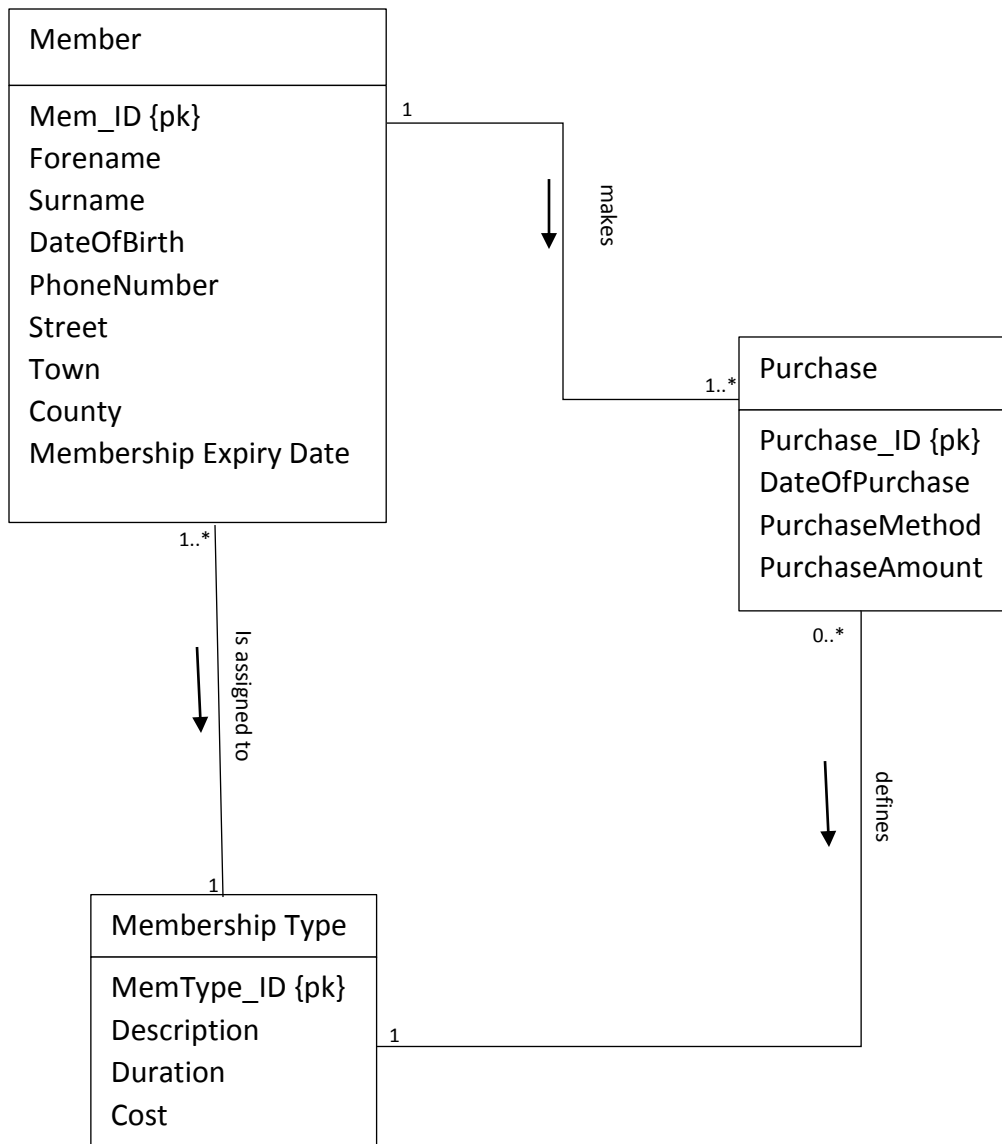


## Level 2 DFD:

P4 Transaction Processes



## 5. Data Model



Assuming a Member can only be assigned to one Membership Type at a time

Assuming a Purchase is for only one Membership Type

## 5.1 Relational Model

Members (Mem\_ID, Forename, Surname, DateOfBirth, PhoneNumber, Street, Town, County, Membership\_ExpiryDate, MemType\_ID)

Purchases (Purchase\_ID, DateOfPurchase, PurchaseAmount, PurchaseMethod, Member\_ID, MemType\_ID)

Membership Types (MemType\_ID, Description, Duration, Cost)

## 6. Database Schema

**Schema:** GymSYS

**Relation:** Members

Attributes:

Mem\_ID numeric (4)  
Forename char (15)  
Surname char (15)  
DateOfBirth date  
PhoneNumber numeric (11)  
Street char (20)  
Town char (20)  
County char (20)  
Membership\_ExpiryDate date  
MemType\_ID numeric (4)

Primary Key: Mem\_ID

Foreign Key: MemType\_ID REFERENCES Membership Types

**Relation:** Purchases

Attributes:

Purchase\_ID numeric (4)  
DateOfPurchase date  
PurchaseAmount numeric (5, 2)  
PurchaseMethod char (20)  
Mem\_ID numeric (4)  
MemType\_ID numeric (4)

Primary Key: Purchase\_ID

Foreign Key: Mem\_ID REFERENCES Members



## MemType\_ID REFERENCES Membership Types

**Relation:** Membership Types

Attributes:

MemType\_ID numeric (4)

Description char (15)

Duration char (15)

Cost numeric (5, 2)

Primary Key: MemType\_ID

## 7. Program Specifications

### 7.1 Create Membership Type

**System :** GymSYS

**Author:** Alan Devlin

**Module :** Create Membership Type

**Date:** 28/11/13

#### Invoked By:

Membership Type Menu

#### Invokes:

None

#### Inputs:

MemType\_ID

Description

Duration

Cost

#### Outputs:

New Membership Type Record

#### Screen Layout:

Create New Membership Type

MemType\_ID

Description

Duration

Cost

## Error Messages:

1. Field must be entered
2. This field may not contain illegal values
3. Please select a Duration option from the drop down menu
4. This field must be numeric

## Process:

BEGIN

Display Form

DO Until EXIT or BACK

If Add is selected

If MemType\_ID is blank then

Do Display Error Message (1)

Exit

Else If MemType\_ID contains illegal values then

Do Display Error Message (2)

Exit

Else If Description is blank then

Do Display Error Message (1)

Exit

Else If Description contains illegal values then

Do Display Error Message (2)

Exit

Else If Duration is blank then

Do Display Error Message (3)

Exit

Else If Cost is blank then

Do Display Error Message (1)

Exit

```
        Else If Cost contains non numeric values then
            Do Display Error Message (4)
            Exit
        If EXIT
            Terminate Program
        If BACK
            Return to Membership Type Menu
    Else
        Open Database
        Write Membership Type Record
        Close Database
        Display confirmation message "Membership Type Created"
        DoClearScreen()
```

**END**

### **Display Error Message(X)**

```
Begin
    IF X =1 then
        Display "This field must be entered"
    ELSE IF X=2 then
        Display "This field may not contain any illegal values"
    ELSE IF X=3 then
        Display "Please select a Duration option from the drop down menu"
    ELSE IF X=4 then
        Display "This field must be numeric"
    END IF
End
```

**ClearScreen()**

Begin

MemType\_ID = ""

Description = ""

Duration = ""

Cost = ""

End

## 7.2 Create New Member

**System :** GymSYS

**Author:** Alan Devlin

**Module :** Create New Member

**Date:** 28/11/13

### Invoked By:

Members Menu

### Invokes:

None

### Inputs:

MemType

Forename

Surname

Phone Number

Street

Town

County

Date of Birth

Payment Details

### Outputs:

New Member Record

## Screen Layout:

The screenshot shows a Windows-style application window titled "Create New Member". The window has a title bar with standard minimize, maximize, and close buttons. Inside the window, the title "Create New Member" is displayed in blue. Below the title, there is a label "Membership\_ID" followed by a text box containing the value "0004". A section titled "Enter Member Details" is enclosed in a light gray border. This section contains several input fields: "MemType" is a dropdown menu; "First Name", "Surname", "Phone Number", "Street", "Town", and "County" are standard text boxes; "Date of Birth" is a date picker showing "10 December 2013"; and "Payment Details" is a text box. Below these fields, within the same bordered area, is a button labeled "Create New Member". Outside this bordered area, at the bottom right of the window, are two buttons: "Back" and "Exit".

## Error Messages:

5. Field must be entered
6. This field may not contain illegal values
7. Please select MemType from the drop down menu
8. The selected date is in the future
9. This field must be numeric
10. The selected date must be in the future

## Process:

BEGIN

Display Form

Do GetNextMembership\_ID()

Do Until EXIT or BACK

If Create New Member is selected

    If MemType is blank then

        Do Display Error Message (3)

        Exit

    Else If Forename is blank then

        Do Display Error Message (1)

        Exit

    Else If Forename contains illegal values then

        Do Display Error Message (2)

        Exit

    Else If Surname is blank then

        Do Display Error Message (1)

        Exit

    Else If Surname contains illegal values then

        Do Display Error Message (2)

        Exit

    Else If Phone Number does not contain numeric values then

        Do Display Error Message (5)

        Exit

    Else If Phone Number is blank then

        Do Display Error Message (1)

        Exit



Else If Street is blank then

Do Display Error Message (1)

Exit

Else If Street contains illegal values then

Do Display Error Message (2)

Exit

Else If Town is blank then

Do Display Error Message (1)

Exit

Else If Town contains illegal values then

Do Display Error Message (2)

Exit

Else If County is blank then

Do Display Error Message (1)

Exit

Else If County contains illegal values then

Do Display Error Message (2)

Exit

Else If Date of Birth is not selected then

Do Display Error Message (1)

Exit

Else If Date of Birth contains a date in the future then

Do Display Error Message (4)

Exit

Else If Payment Details is blank then

Do Display Error Message (1)

Exit

Else If Payment Details contains illegal values then

Do Display Error Message (2)

```
        Exit
    Else If Membership Expiry Date is not selected then
        Do Display Error Message (1)
        Exit
    Set Membership Expiry Data
        System Date + Duration of Membership Type
    Exit
Else
    Open Database
    Write Member Record
    Close Database
    Display Confirmation message "Member registration Complete"
    DoClearScreen()

    IF EXIT
        Terminate Program
    IF BACK
        Return to Members Menu
```

**END**

### **Display Error Message (X)**

Begin

```
    IF X =1 then
        Display "This field must be entered"
    IF X =2 then
        Display "This field may not contain illegal values"
    IF X =3 then
        Display "Please select MemType from the drop down menu"
```

IF X =4 then

Display "The selected date is in the future"

IF X =5 then

Display "This field must be numeric"

IF X =6 then

Display "The selected date must be in the future"

END IF

End

### **GetNextMembership\_ID()**

Begin

Connect to database

Retrieve biggest Membership\_ID assigned from Members Files

IF Last Membership\_ID is NULL

Set Membership\_ID = 0001

ELSE

Set Membership\_ID = Biggest Membership\_ID + 1

Close Database

End

### **ClearScreen()**

Begin

DO GetNextMembership\_ID

Forename = ""

Surname = ""

Phone Number = ""

Street = ""

Town = ""

County = ""

Date of Birth = ""

Payment Details = ""

End

### 7.3 Edit Member Details

**System :** GymSYS

**Author:** Alan Devlin

**Module :** Edit Member Details

**Date:** 28/11/13

#### Invoked By:

Members Menu

#### Invokes:

None

#### Inputs:

Forename

Surname

Phone Number

Street

Town

County

Date of Birth

Payment Details

#### Outputs:

Update Member Details

**Screen Layout:**

Edit Members

Edit Member Details

Membership\_ID0004

Edit Member Details

First NameFrank

SurnameMurphy

Phone Number086-34512390

Street67 Ashley Downs

TownTralee

CountyCo. Kerry

Date of Birth10 December 2013

Payment DetailsCheque

Edit Member Details

Back

Exit

## Error Messages:

11. Field must be entered
12. This field may not contain illegal values
13. Please select MemType from the drop down menu
14. The selected date is in the future
15. This field must be numeric
16. The selected date must be in the future

## Process:

BEGIN

Display Form

Do Until EXIT or BACK

    If Edit Member Details

        If Forename is blank then

            Do Display Error Message (1)

            Exit

        Else If Forename contains illegal values then

            Do Display Error Message (2)

            Exit

        Else If Surname is blank then

            Do Display Error Message (1)

            Exit

        Else If Surname contains illegal values then

            Do Display Error Message (2)

            Exit

        Else If Phone Number does not contain numeric values then

            Do Display Error Message (5)

Exit

Else If Phone Number is blank then

Do Display Error Message (1)

Exit

Else If Street is blank then

Do Display Error Message (1)

Exit

Else If Street contains illegal values then

Do Display Error Message (2)

Exit

Else If Town is blank then

Do Display Error Message (1)

Exit

Else If Town contains illegal values then

Do Display Error Message (2)

Exit

Else If County is blank then

Do Display Error Message (1)

Exit

Else If County contains illegal values then

Do Display Error Message (2)

Exit

Else If Date of Birth is not selected then

Do Display Error Message (1)

Exit

Else If Date of Birth contains a date in the future then

Do Display Error Message (4)

Exit

Else If Payment Details is blank then



```
        Do Display Error Message (1)
        Exit
    Else If Payment Details contains illegal values then
        Do Display Error Message (2)
        Exit
    Else
        Open Database
        Update Member Record
        Close Database
        DoClearScreen()
    IF EXIT
        Terminate Program
    IF BACK
        Return to Members Menu
```

**END**

### **Display Error Message (X)**

```
Begin
    IF X =1 then
        Display "This field must be entered"
    IF X =2 then
        Display "This field may not contain illegal values"
    IF X =3 then
        Display "Please select MemType from the drop down menu"
    IF X =4 then
        Display "The selected date is in the future"
    IF X =5 then
        Display "This field must be numeric"
```

IF X =6 then

    Display "The selected date must be in the future"

END IF

End

**ClearScreen()**

Begin

    Forename = ""

    Surname = ""

    Phone Number = ""

    Street = ""

    Town = ""

    County = ""

    Date of Birth = ""

    Payment Details = ""

End

## 8. Conclusion

The overall objective of this project was to create specifications that can be used by a software developer to create GymSYS. The project contains specific user and system requirements which describe the functions that the system must be able to perform. Data flow Diagrams were also included to show the interaction between each process and where they would be stored. A data model and databases schema was included to show how each process would be implemented in a database.

A prototype and program specifications were included to give an indication of how the software would look, and the interactions taking place behind the user interface. It was important to make the prototype as user friendly as possible. The prototype also included the functionality to be able to tell the user if they had not entered a suitable attribute when filling in a text field.

An extra function that could be added to the system in the future would be to store notes about each member. This would be particularly beneficial to keep a record of any fitness goals and achievements they have achieved. The member's personal trainer could then view the members profile and see what work is needed to be done.

Although it deals with quite basic processes, I think that GymSYS could be implemented in a real world system. With some added functionality I think it can achieve its primary goal of being able to register a member and allocate them to their relevant membership type.