

POLITECHNIKA OPOLSKA

FACULTY OF ELECTRICAL ENGINEERING AUTOMATIC CONTROL AND INFORMATICS

Development of a Sport center application with WINdev



Final Project

David Alonso Ayala

Victor Beltrán López

Director

Krzysztof Bartecki

Co-director

Juan Carlos Ruíz García

INDEX

Index.....	3
1. Problem statement.....	4
2. Functional requirements.....	5
2.1 Functional requirements list.....	5
2.2 Functional Decomposition Diagram.....	8
2.3 Partial Event Decomposition Diagram.....	9
3. Non-Functional requirements.....	10
4. Process Model (DFD)	11
4.1 Context Process Model (Context DFD).....	11
4.2 System DFD (0 Level DFD).....	12
4.3 Lower-level diagrams.....	13
5. Data Model (ERD).....	21
5.1 Context Data Model (Context ERD)	21
5.2 Complete Data Model (Complete ERD)	22
5.3 Data dictionary.....	23
6. Physical Data Model (PDM)	30
6.2 SQL Anywhere Database generated from PDM.....	30
7. Object oriented model.....	39
7.1 Use Case Diagram	39
7.2 Class Diagram.....	44
7.3 Sequence diagrams	45
7.4 Class declarations code generated from the class diagram.....	48

1. Problem statement

The proposed Final Project is part of the suite of applications for business management. This project will develop while we are at Erasmus period in Poland (Politechnika Opolska), so we take aim to improve the management of such centers in this country, since in many of these centers there is no computerized system. Trying to computerize this kind of centers has major advantages both for the users of the center and for workers and the organization's own site.

This project arises because we have seen the functioning of such organizations in both the private and the university and we felt we have a large degree of improvement.

To achieve the ultimate goal, we have defined a script to follow. First, before you start programming, we will make a list of functional and non functional requirements, develop a flow diagram (DFD) and model database. Once all this, we can begin to develop what will be the final application.

In developing the project we will use the development tool Windev. The following outlines the tools and how we will carry out the project.

- **Windev**

Windev, is an integrated development environment has been created by a French company PCSoft. The language used for programming Windev W-Language, is a 4th generation language, easy to use and very fast to program with it. WinDev has a GUI editor for creating graphical user interfaces using drag/drop.

- **Database.**

Our application will use a database to store all data of interest. The database engine will Windev provided to us, his name is **HiperFile**.

- **Microsoft Visio**

Before we start to develop our project, we will go through the design stage. At this stage, we will create the Data Flow Diagram (DFD). A DFD is simply a graphical representation of the "flow" of data through an information system. or another stage in the design phase will be to create the Unified Modeling Language (UML). A UML is a graphical language for visualizing, specifying, constructing and documenting a system. o To develop both DFD and UML use software Microsoft Visio that is for free for students in [MSDN Academic Alliance](#).

2. Functional requirements

2.1. Functional requirements list

Requirement	Explanation
Author :	Client
Requirement Number :	SCA - 1.0
Requirement Title :	Sign up activities
Requirement Text :	The system should allow the user to register the activities on offer
Requirement Type:	Functional

Requirement	Explanation
Author :	Client
Requirement Number :	SCA - 2.0
Requirement Title :	Reserve track/court
Requirement Text :	The system should allow the user to book a track or sports area that is available
Requirement Type:	Functional

Requirement	Explanation
Author :	Client
Requirement Number :	SCA - 3.0
Requirement Title :	Modify personal data (low level)
Requirement Text :	The system should allow the user to modify personal data like "password" for login in to the system.
Requirement Type:	Functional

Requirement	Explanation
Author :	Employer
Requirement Number :	SCA - 4.0
Requirement Title :	Activities info
Requirement Text :	The system should allow the employer see which activities are under its responsibility
Requirement Type:	Functional

Requirement	Explanation
Author :	Employer
Requirement Number :	SCA - 5.0
Requirement Title :	Timetable info
Requirement Text :	The system should allow the employer see his timetable of the actual week .
Requirement Type:	Functional

Requirement	Explanation
Author :	Department Manager
Requirement Number :	SCA - 6.0
Requirement Title :	Timetable management
Requirement Text :	The system should allow the department manager generate the timetable for all employers under his responsibility.
Requirement Type:	Functional

Requirement	Explanation
Author :	Department Manager
Requirement Number :	SCA - 7.0
Requirement Title :	Orders management
Requirement Text :	The system should allow the department manager make new orders of equipment for the sport areas.
Requirement Type:	Functional

Requirement	Explanation
Author :	Department Manager
Requirement Number :	SCA - 8.0
Requirement Title :	Activities management
Requirement Text :	The system should allow the department manager make new activities or modify already existing.
Requirement Type:	Functional

Requirement	Explanation
Author :	Administration
Requirement Number :	SCA - 9.0
Requirement Title :	Staff / Users Management
Requirement Text :	The system should allow the administration to hire/dismiss employers and add/delete users
Requirement Type:	Functional

Requirement	Explanation
Author :	Administration
Requirement Number :	SCA - 9.0
Requirement Title :	Economy Management
Requirement Text :	The system should allow the administration to generate entry and expenditures.
Requirement Type:	Functional

2.2. Functional Decomposition Diagram

The Functional Decomposition Diagram shows Sport center as the root of the system. It also displays the five main subsystems related to our project: Reserves, activities, economy, personal and area's process. Below each parent subsystem is the processes that it handles, such as make new reserve, auditory or personal management for example.

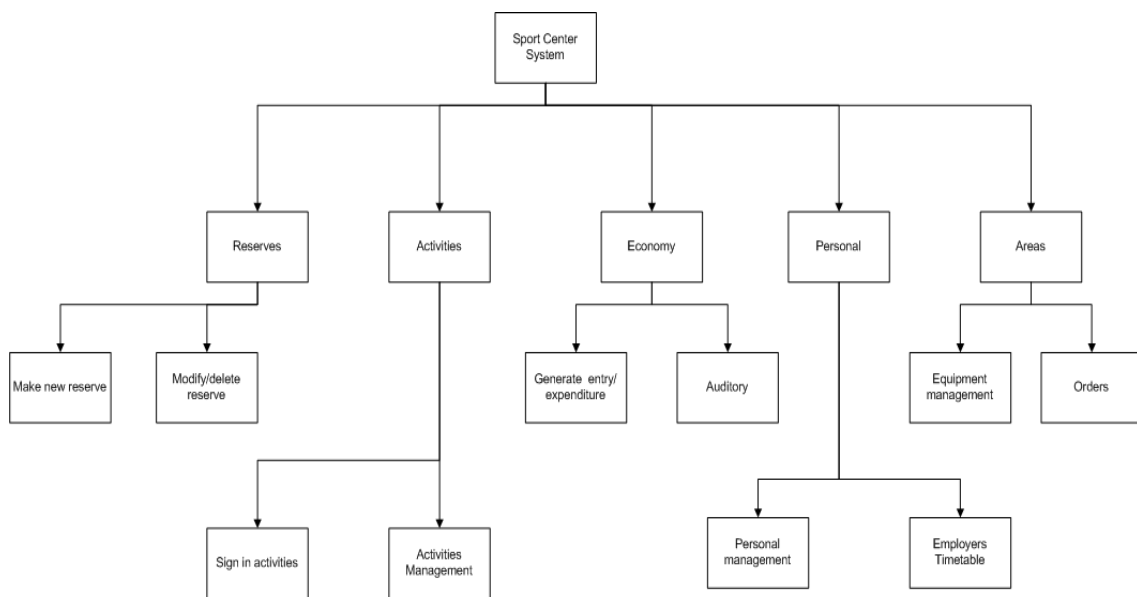


Figure 1 - Functional Decomposition Diagram

2.3. Partial Event Decomposition Diagram

The following Diagram describes the system in greater detail, describing the tasks performed in each process. For example, the personal Management process includes the tasks of add/delete new client/employer and modify employers/client data.

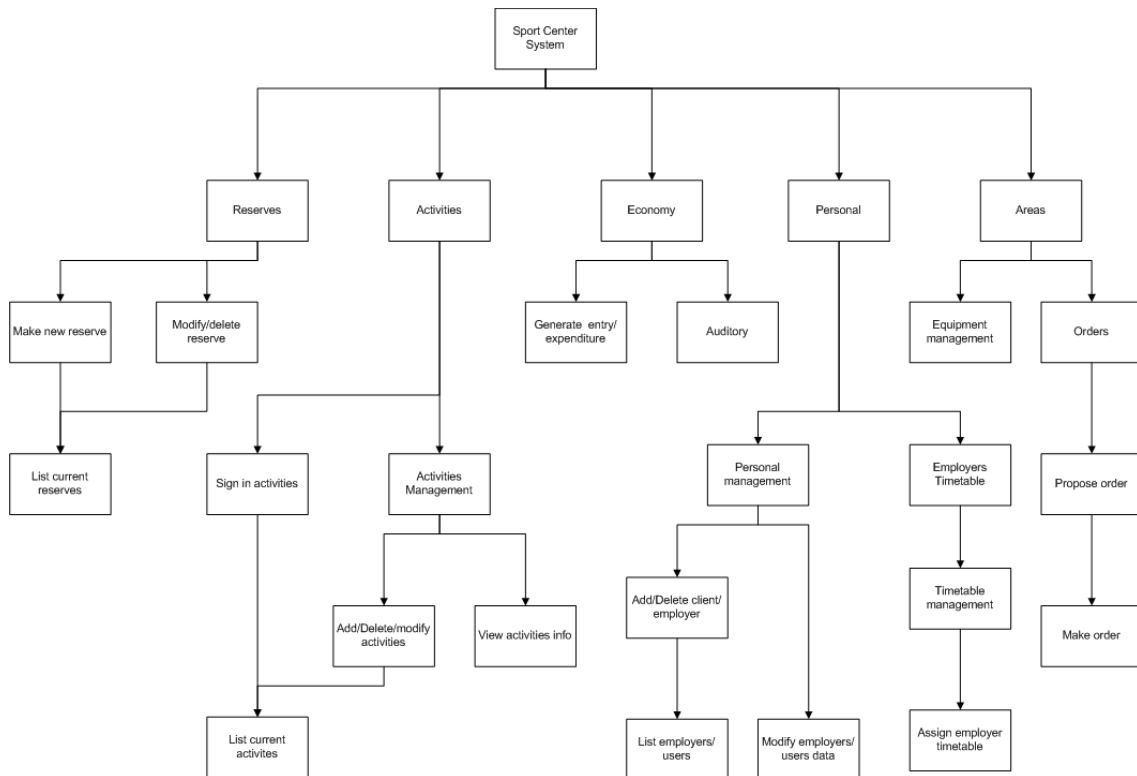


Figure 2 – Partial event Decomposition Diagram

3. Non-Functional requirements

- **Hardware Interface.**

The system can be managed from any personal computer. It doesn't need any special peripheral or extra protocol. That computer should be in the sport center enclosure.

- **Software Interface.**

This application is thinking for run in Windows operative system. It does not need any particular application for start to run, the only one thing that the user must to do is install our application with the installer that we give them.

4. Process Model (DFD)

4.1. Context Diagram

The following diagram shows the Sport Center System Context Diagram. The context diagram defines the scope and boundary for the system and the project.

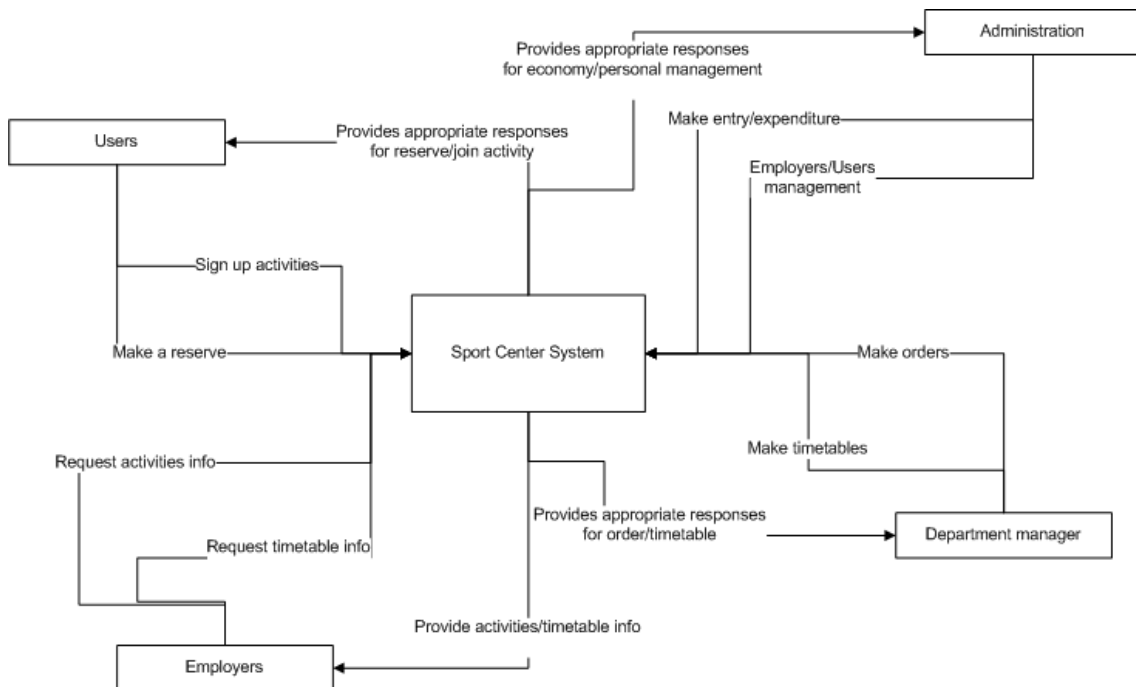


Figure 3 – Context Diagram

4.2 System DFD

The System Diagram for Sport Center System contains one principal processes/events and there are five external agents: Users(clients), employers, department manager, administration and super user (root).

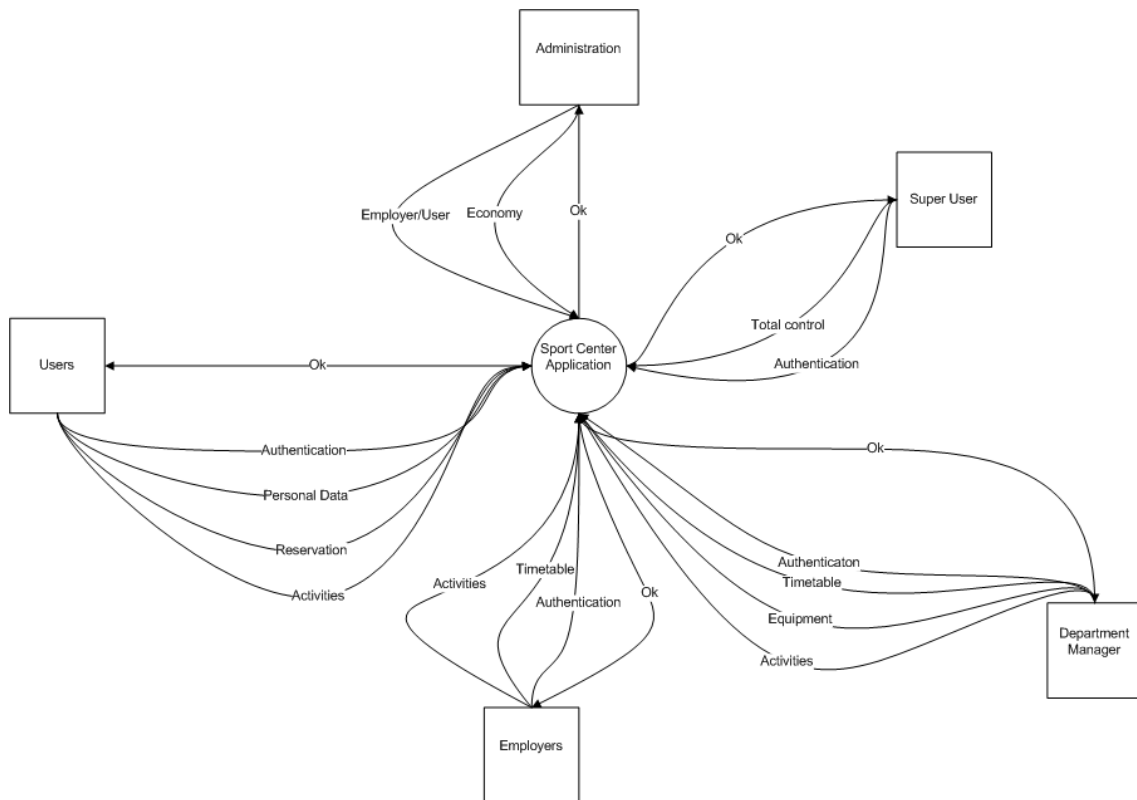


Figure 4 – System dfd

4.3. Lower level diagrams

4.3.1. DFD level 1

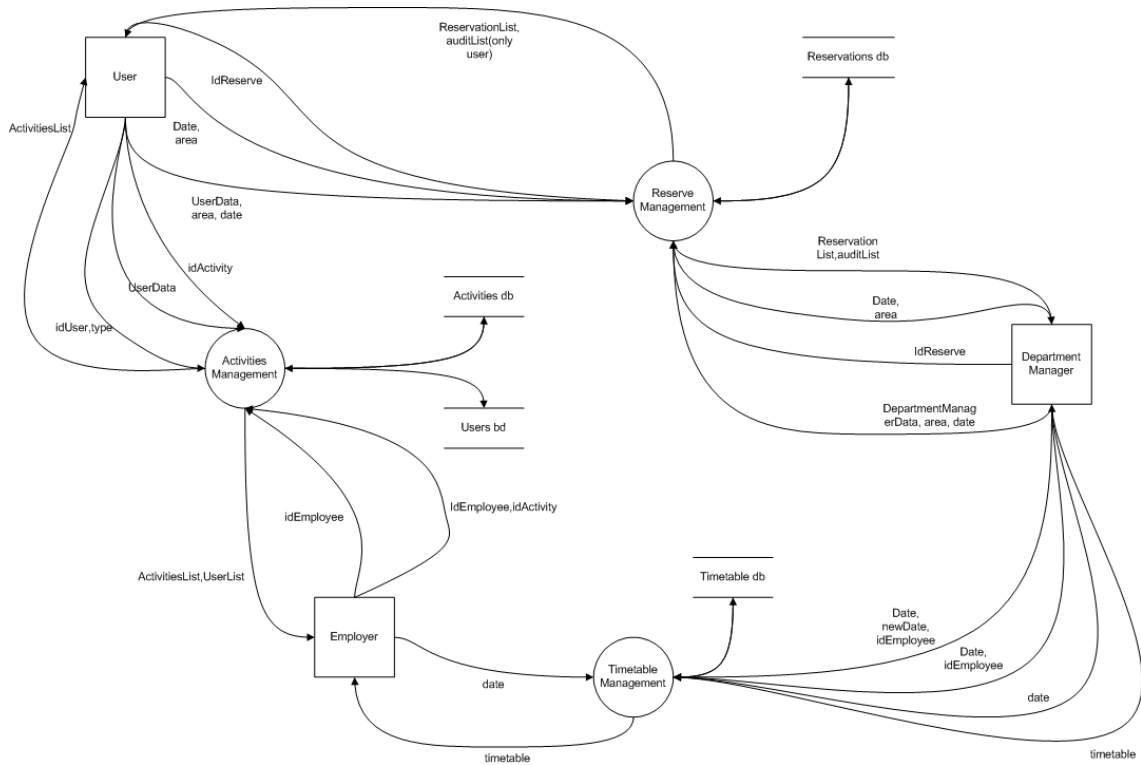


Figure 5 – dfd level 1

4.3.2. DFD level 2

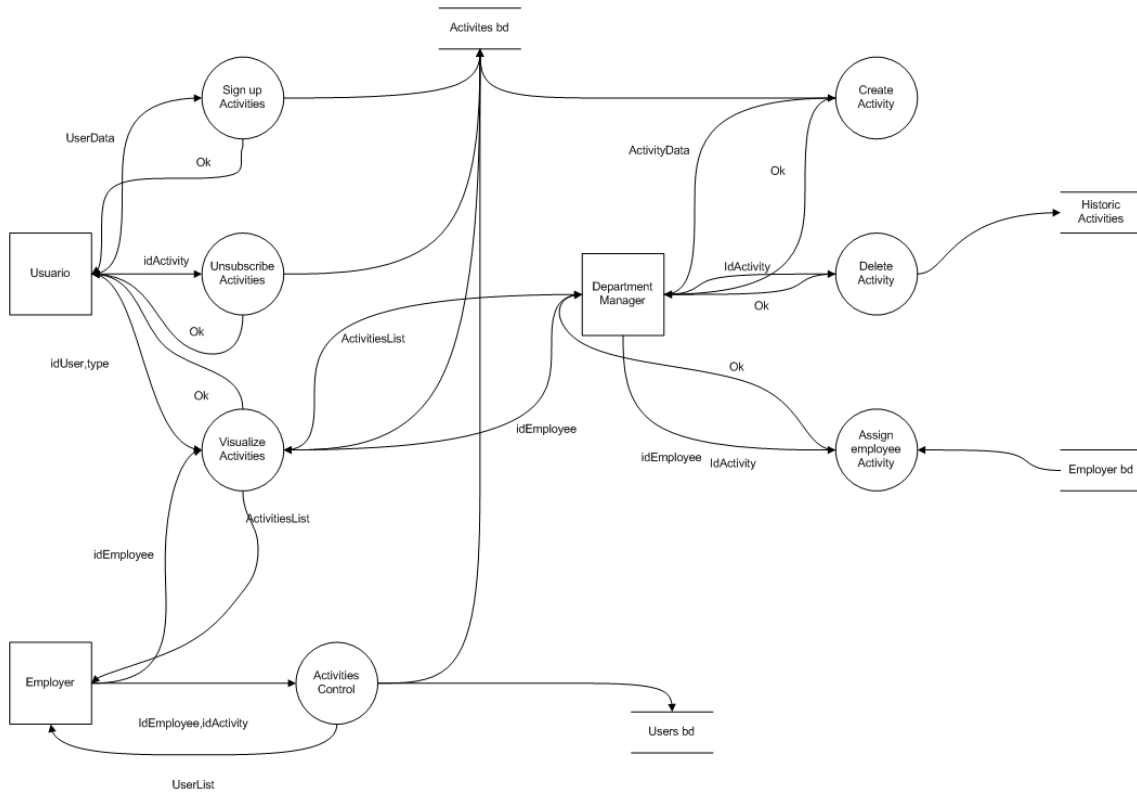


Figure 7 – activities dfd level 2

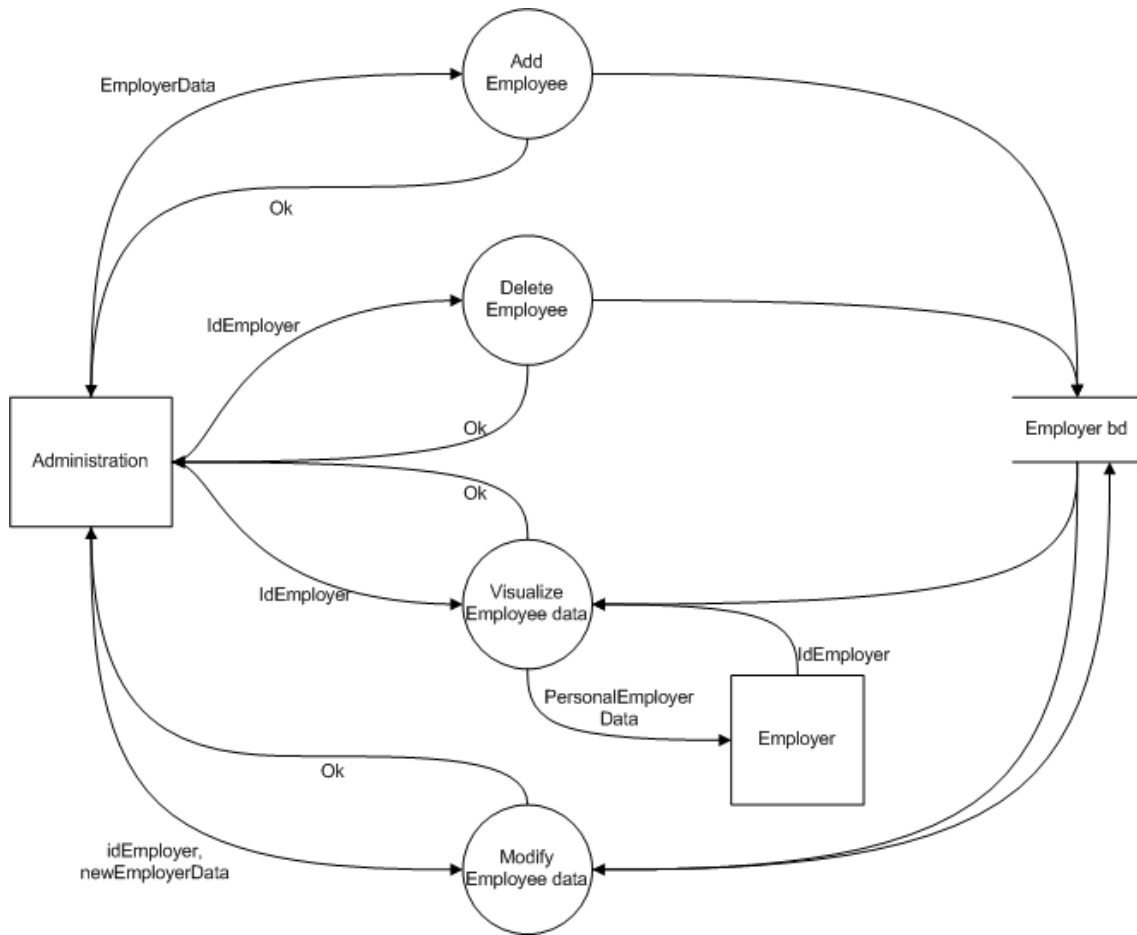


Figure 8 – administration dfd level 2

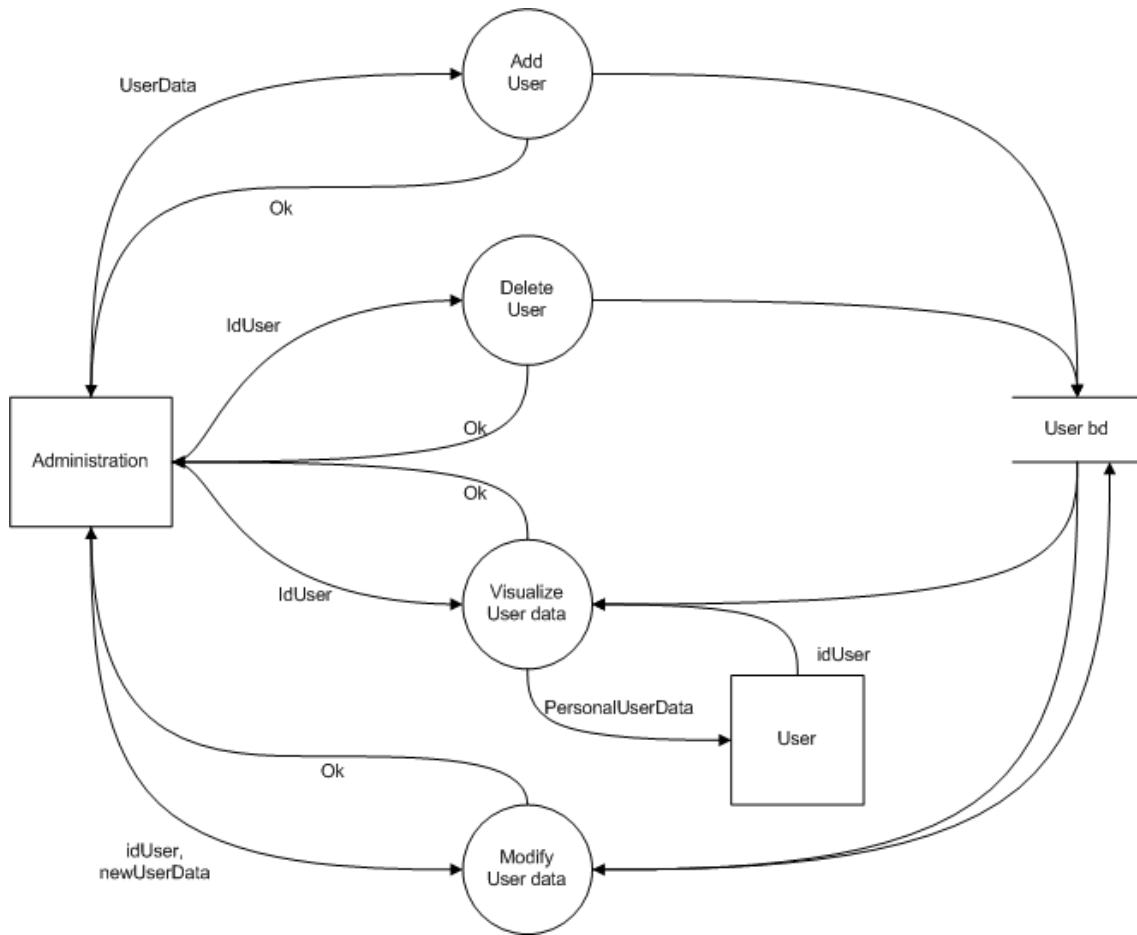


Figure 9 – administration dfd level 2

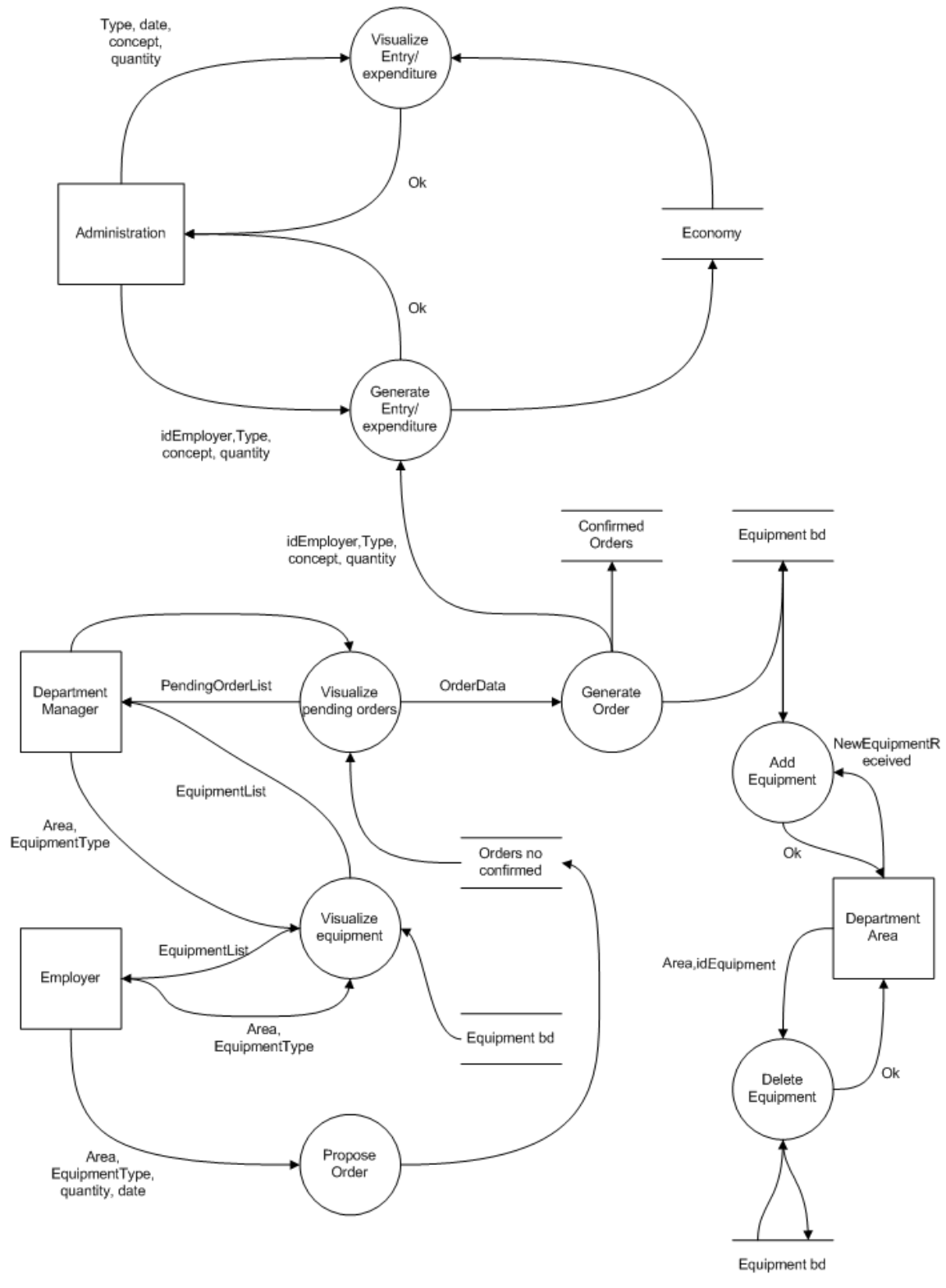


Figure 10 – administration dfd level 2

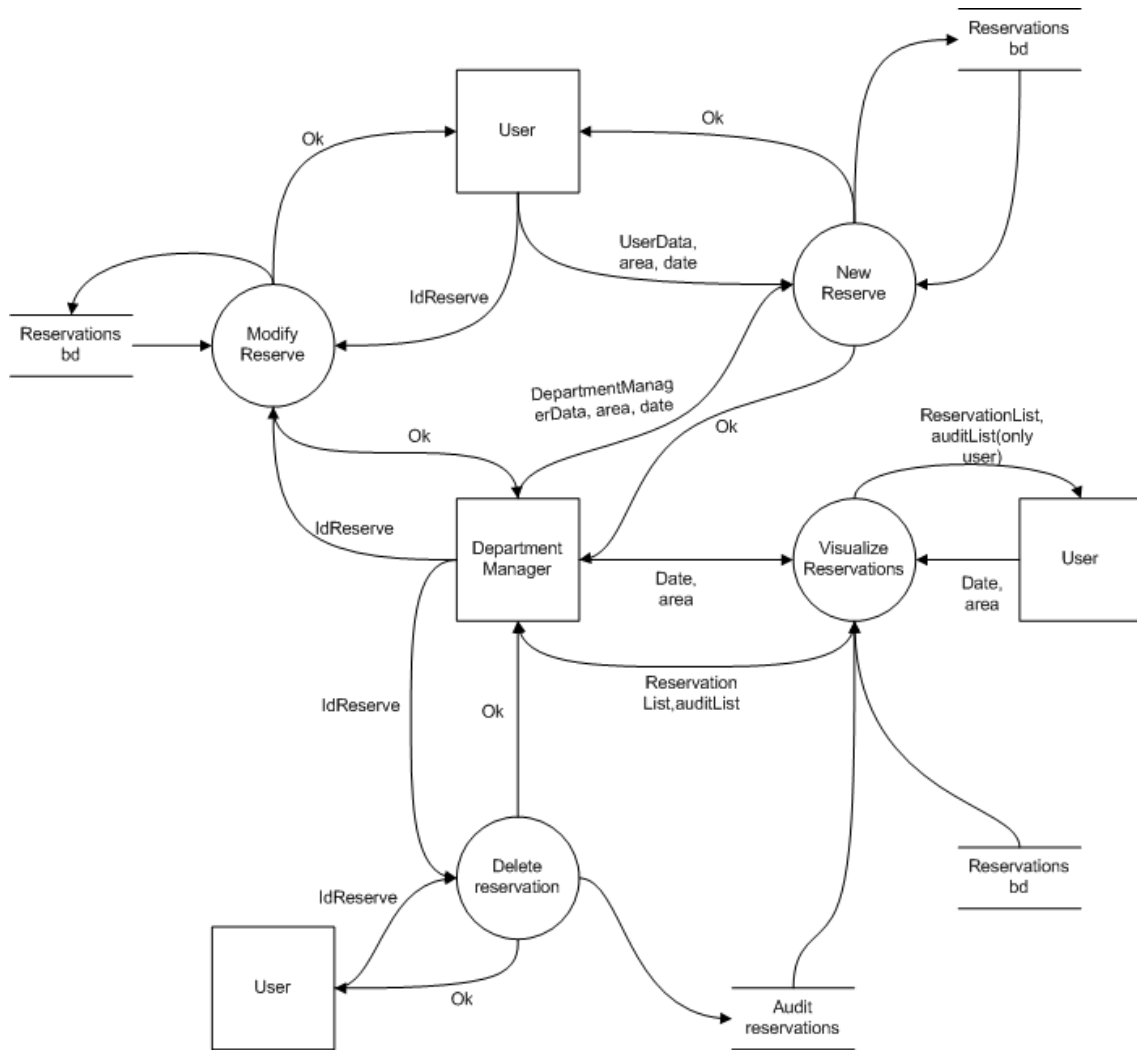


Figure 11– reserves dfd level 2

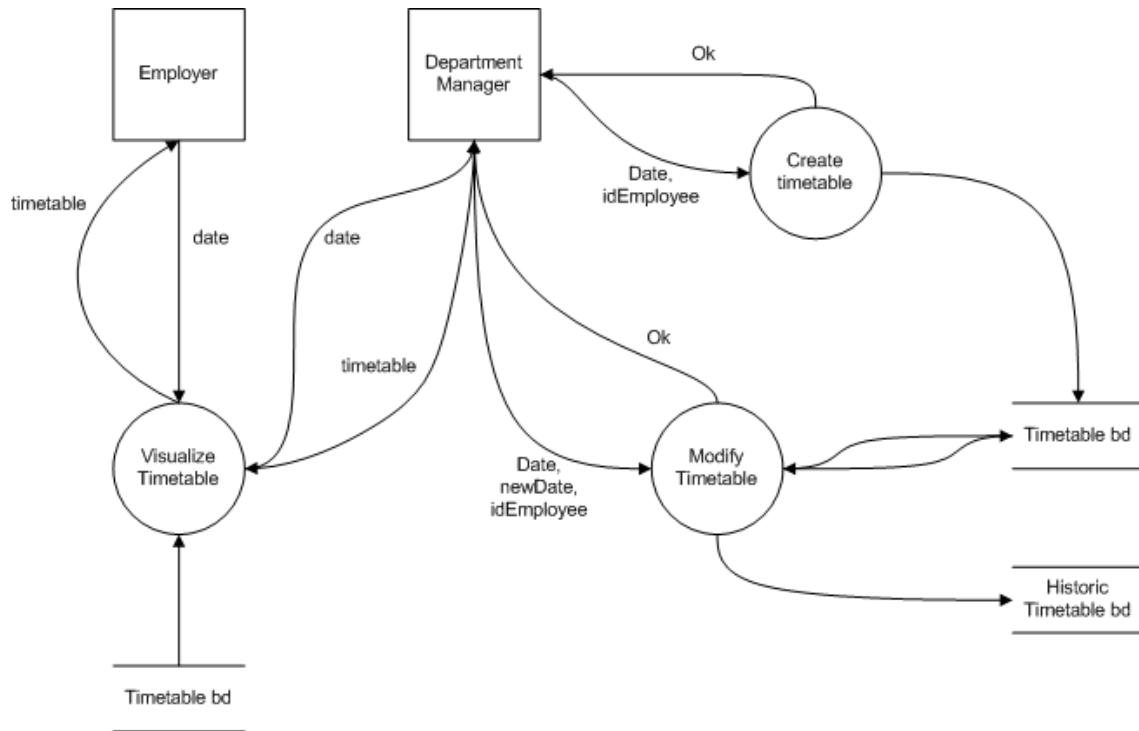


Figure 12 – timetable dfd level 2

5. Data Model (ERD)

5.1. Context Data Model

The Context Data Model for Sport Center system consists of the following entities: Client, Employer, Economy, Equipment , Reserves, Activities, Orders and their respective relational tables..

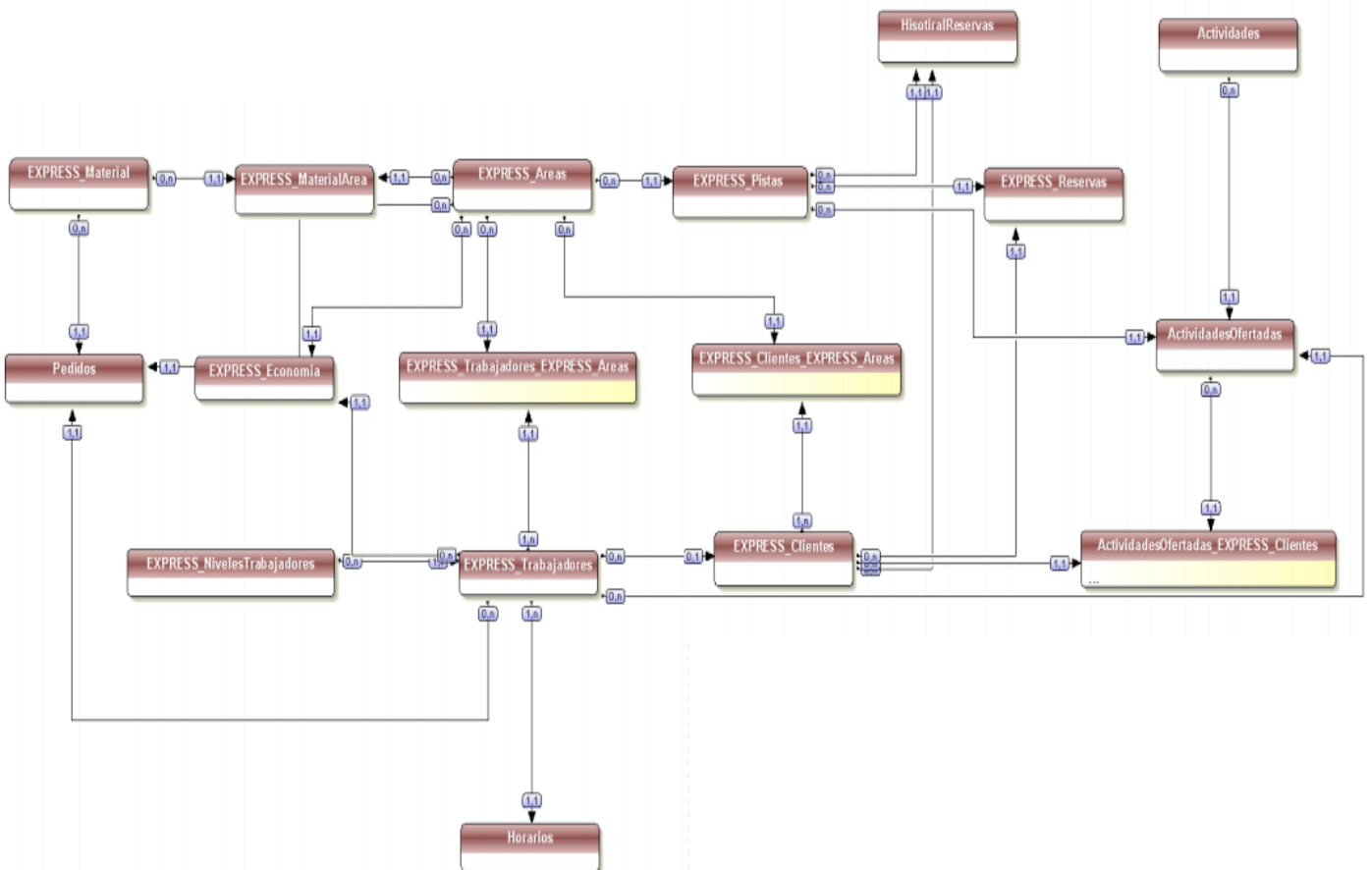


Figure 13 - Context Data Model

5.2. Complete Data Model

The Complete Data Model for Sport Center system consists of the same entities listed in the Context Data Model. However, this model contains all of the attributes associated with each entity. The following are the business rules for establishing each of the relationships:

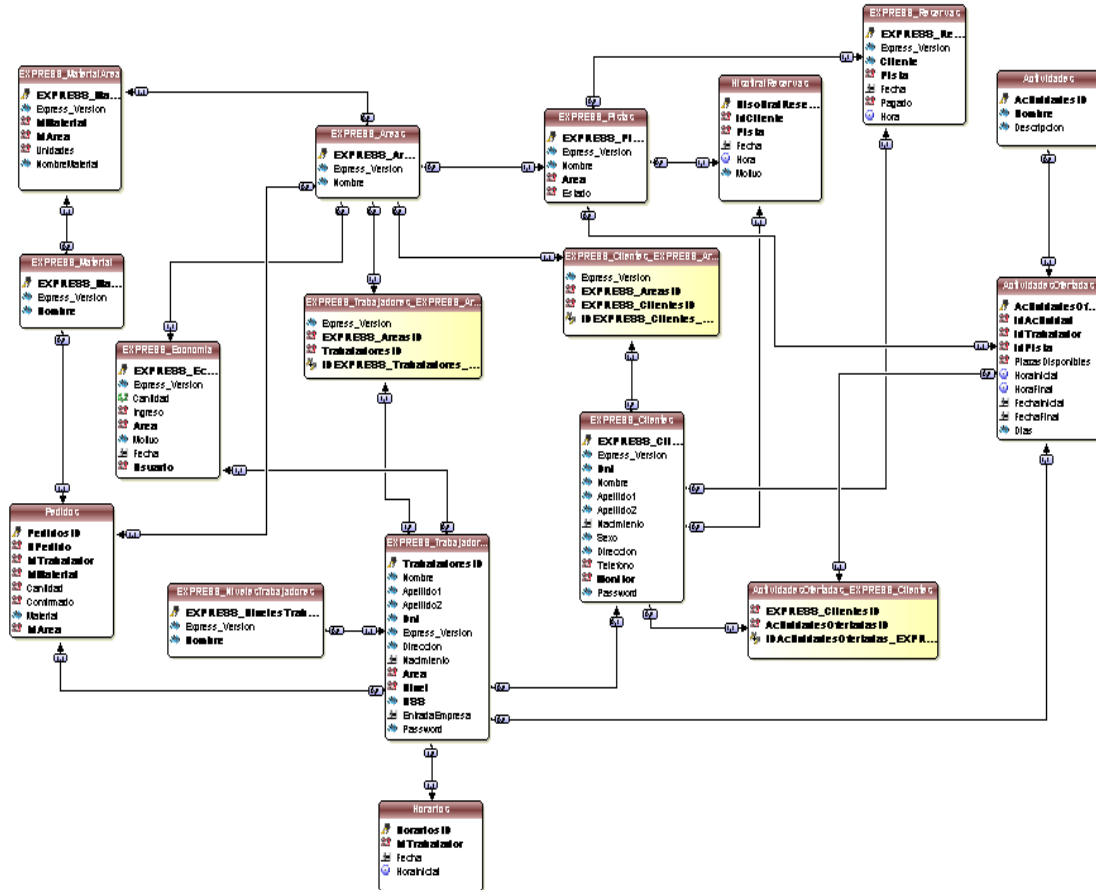


Figure 14 - Complete Data Model

5.3. Data dictionary

Item	Type	Size	Unique	Key	Used by...
ActividadesID	Automatic Identifier (4 bytes)				Actividades
ActividadesOfertadasID	4-byte integer				ActividadesOfertadas_EXPRESS_Cientes
	Automatic Identifier (4 bytes)				ActividadesOfertadas
Apellido1	String	100			EXPRESS_Cientes
	String	100			EXPRESS_Trabajadores
	String	100			HistoricoCientes
Apellido2	String	100			EXPRESS_Cientes
	String	100			EXPRESS_Trabajadores
	String	100			HistoricoCientes
Area	4-byte integer				EXPRESS_Pistas
	4-byte integer				EXPRESS_Economia
	4-byte integer				EXPRESS_Trabajadores
Cantidad	4-byte real				EXPRESS_Economia
	4-byte integer				Pedidos

Item	Type	Size	Unique	Key	Used by...
Carnet					<Unused>
Cliente	String	9			EXPRESS_Reservas
Confirmado	Boolean				Pedidos
Descripcion	String	200			Actividades
Dias	String	50			ActividadesOfertadas
Direccion	String	200			EXPRESS_Clientes
	String	200			EXPRESS_Trabajadores
	String	200			HistoricoClientes
Dni	String	9			EXPRESS_Clientes
	String	9			EXPRESS_Trabajadores
	String	9			HistoricoClientes
EntradaEmpresa	Date (yyyymmdd)				EXPRESS_Trabajadores
Estado	Boolean				EXPRESS_Pistas
EXPRESS_AreasID	4-byte integer				EXPRESS_Clientes_EXPRESS_Areas
	4-byte integer				EXPRESS_Trabajadores_EXPRESS_Areas
	Automatic Identifier (4 bytes)				EXPRESS_Areas
EXPRESS_ClientesID	4-byte integer				ActividadesOfertadas_EXPRESS_Clientes
	4-byte integer				EXPRESS_Clientes_EXPRESS_Areas
	Automatic Identifier (4 bytes)				EXPRESS_Clientes
EXPRESS_EconomialID	Automatic Identifier (4 bytes)				EXPRESS_Economia
EXPRESS_MaterialAreaID	Automatic Identifier (4 bytes)				EXPRESS_MaterialArea
EXPRESS_MaterialID	Automatic Identifier (4 bytes)				EXPRESS_Material
EXPRESS_NivelesTrabajadoresID	Automatic Identifier (4 bytes)				EXPRESS_NivelesTrabajadores
Express_Nombre					<Unused>
EXPRESS_PistasID	Automatic Identifier (4 bytes)				EXPRESS_Pistas
EXPRESS_ReservasID	Automatic Identifier (4 bytes)				EXPRESS_Reservas
EXPRESS_TrabajadoresID					<Unused>
Express_Version	String	15			EXPRESS_Pistas
	String	15			EXPRESS_Reservas
	String	15			EXPRESS_Economia
	String	15			EXPRESS_MaterialArea
	String	15			EXPRESS_Material
	String	15			EXPRESS_Clientes_EXPRESS_Areas

String	15			EXPRESS_Trabajadores_EXPRESS_Areas
--------	----	--	--	------------------------------------

Item	Type	Size	Unique	Key	Used by...
	String	15			EXPRESS_NivelesTrabajadores
	String	15			EXPRESS_Areas
	String	15			EXPRESS_Clientes
	String	15			EXPRESS_Trabajadores
Fecha	Date (yyyymmdd)				HisotiralReservas
	Date (yyyymmdd)				EXPRESS_Reservas
	Date (yyyymmdd)				EXPRESS_Economia
	Date (yyyymmdd)				Horarios
FechaFinal	Date (yyyymmdd)				ActividadesOfertadas
FechaInicial	Date (yyyymmdd)				ActividadesOfertadas
HisotiralReservasID	Automatic Identifier (4 bytes)				HisotiralReservas
HistoricoClientesID	Automatic Identifier (4 bytes)				HistoricoClientes
Hora	Time (hhmm)	4			HisotiralReservas
	Time (hhmm)	4			EXPRESS_Reservas
HoraFinal	Time (hhmm)	4			ActividadesOfertadas
Horainicial	Time (hhmm)	4			ActividadesOfertadas
	Time (hhmm)	4			Horarios
HorariosID	Automatic Identifier (4 bytes)				Horarios
idActividad	4-byte integer				ActividadesOfertadas
IDActividadesOfertadas_EXPRESS_Clientes	Composite Key	8			ActividadesOfertadas_EXPRESS_Clientes
IdArea	4-byte integer				EXPRESS_MaterialArea
	4-byte integer				Pedidos
idCliente	4-byte integer				HisotiralReservas
IDEXPRESS_Clientes_EXPRESS_Areas	Composite Key	8			EXPRESS_Clientes_EXPRESS_Areas
IDEXPRESS_Trabajadores_EXPRESS_Areas	Composite Key	8			EXPRESS_Trabajadores_EXPRESS_Areas
IdMaterial	4-byte integer				EXPRESS_MaterialArea
	4-byte integer				Pedidos
idPista	4-byte integer				ActividadesOfertadas
idTrabajador	4-byte integer				ActividadesOfertadas
	4-byte integer				Horarios
	4-byte integer				Pedidos
Ingreso	Boolean				EXPRESS_Economia

Material	String	100			Pedidos
Monitor	4-byte integer				EXPRESS_Cientes

Item	Type	Size	Unique	Key	Used by...
Motivo	String	200			HisotiralReservas
	String	200			EXPRESS_Economia
Nacimiento	Date (yyyymmdd)				EXPRESS_Clientes
	Date (yyyymmdd)				EXPRESS_Trabajadores
	Date (yyyymmdd)				HistoricoClientes
Nivel	4-byte integer				EXPRESS_Trabajadores
Nombre	String	100			Actividades
	String	100			EXPRESS_Pistas
	String	100			EXPRESS_Material
	String	100			EXPRESS_NivelesTrabajadores
	String	100			EXPRESS_Areas
	String	100			EXPRESS_Clientes
	String	100			EXPRESS_Trabajadores
	String	100			HistoricoClientes
NombreMaterial	String	100			EXPRESS_MaterialArea
NPedido	4-byte integer				Pedidos
NSS	String	50			EXPRESS_Trabajadores
Pagado	Boolean				EXPRESS_Reservas
Password	String	50			EXPRESS_Clientes
	String	50			EXPRESS_Trabajadores
PedidosID	Automatic Identifier (4 bytes)				Pedidos
Pista	4-byte integer				HisotiralReservas
	4-byte integer				EXPRESS_Reservas
PlazasDisponibles	4-byte integer				ActividadesOfertadas
Sexo	String	1			EXPRESS_Clientes
Telefono	4-byte unsigned integer				EXPRESS_Clientes
	4-byte unsigned integer				HistoricoClientes
TrabajadoresID	4-byte integer				EXPRESS_Trabajadores_EXPRESS_Areas

	Automatic Identifier (4 bytes)				EXPRESS_Trabajadores
Unidades	4-byte integer				EXPRESS_MaterialArea
Usuario	4-byte integer				EXPRESS_Economia

6. Physical Data Model (PDM)

-- Script generated by WinDev

-- Tables of PFC_DataBase.wda analysis

-- for Generic SQL (ANSI 92)

-- Creating the Actividades table

```
CREATE TABLE "Actividades" (  
    "ActividadesID" INTEGER PRIMARY KEY ,  
    "Nombre" VARCHAR(100) NOT NULL UNIQUE ,  
    "Descripcion" VARCHAR(200) NOT NULL );
```

-- Creating the ActividadesOfertadas table

```
CREATE TABLE "ActividadesOfertadas" (  
    "ActividadesOfertadasID" INTEGER PRIMARY KEY ,  
    "idActividad" INTEGER NOT NULL ,  
    "idTrabajador" INTEGER NOT NULL ,  
    "PlazasDisponibles" INTEGER NOT NULL ,  
    "HoraInicial" TIME NOT NULL ,  
    "HoraFinal" TIME NOT NULL ,  
    "idPista" INTEGER NOT NULL ,  
    "FechaInicial" DATE NOT NULL ,  
    "FechaFinal" DATE NOT NULL ,  
    "Dias" VARCHAR(50) NOT NULL );
```

```
CREATE INDEX "WDIDX_ActividadesOfertadas_idActividad" ON  
"ActividadesOfertadas" ("idActividad");
```

```
CREATE INDEX "WDIDX_ActividadesOfertadas_idTrabajador" ON  
"ActividadesOfertadas" ("idTrabajador");
```

```
CREATE INDEX "WDIDX_ActividadesOfertadas_idPista" ON "ActividadesOfertadas"  
("idPista");
```

-- Creating the ActividadesOfertadas_EXPRESS_Clientes table

```
CREATE TABLE "ActividadesOfertadas_EXPRESS_Clientes" (
```

```
    "EXPRESS_ClientesID" INTEGER NOT NULL ,
```

```
    "ActividadesOfertadasID" INTEGER NOT NULL );
```

```
CREATE INDEX
```

```
"WDIDX_ActividadesOfertadas_EXPRESS_Clientes_EXPRESS_ClientesID" ON
```

```
"ActividadesOfertadas_EXPRESS_Clientes" ("EXPRESS_ClientesID");
```

```
CREATE INDEX
```

```
"WDIDX_ActividadesOfertadas_EXPRESS_Clientes_ActividadesOfertadasID" ON
```

```
"ActividadesOfertadas_EXPRESS_Clientes" ("ActividadesOfertadasID");
```

-- Creating the EXPRESS_Areas table

```
CREATE TABLE "EXPRESS_Areas" (
```

```
    "EXPRESS_AreasID" INTEGER PRIMARY KEY,
```

```
    "Express_Version" VARCHAR(15) NOT NULL ,
```

```
    "Nombre" VARCHAR(100) NOT NULL );
```

-- Creating the EXPRESS_Clientes table

```
CREATE TABLE "EXPRESS_Clientes" (
```

```
    "EXPRESS_ClientesID" INTEGER PRIMARY KEY,
```

```
    "Express_Version" VARCHAR(15) NOT NULL ,
```

```
    "Nombre" VARCHAR(100) NOT NULL ,
```

```
    "Dni" VARCHAR(9) NOT NULL UNIQUE ,
```

```
    "Apellido1" VARCHAR(100) NOT NULL ,
```

```
    "Apellido2" VARCHAR(100) NOT NULL ,
```

```
    "Nacimiento" DATE NOT NULL ,
```

```
    "Sexo" VARCHAR(1) NOT NULL ,
```

```
    "Direccion" VARCHAR(200) NOT NULL ,
```

```
    "Telefono" NUMERIC(10,0) NOT NULL ,
```

```
    "Monitor" INTEGER NOT NULL ,
```

```
    "Password" VARCHAR(50) NOT NULL );
```

```
CREATE INDEX "WDIDX_EXPRESS_Clientes_Monitor" ON "EXPRESS_Clientes"
("Monitor");
```

-- Creating the EXPRESS_Clientes_EXPRESS_Areas table

```
CREATE TABLE "EXPRESS_Clientes_EXPRESS_Areas" (
    "Express_Version" VARCHAR(15) NOT NULL ,
    "EXPRESS_AreasID" INTEGER NOT NULL ,
    "EXPRESS_ClientesID" INTEGER NOT NULL );
```

```
CREATE INDEX "WDIDX_EXPRESS_Clientes_EXPRESS_Areas_EXPRESS_AreasID" ON
"EXPRESS_Clientes_EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
CREATE INDEX "WDIDX_EXPRESS_Clientes_EXPRESS_Areas_EXPRESS_ClientesID" ON
"EXPRESS_Clientes_EXPRESS_Areas" ("EXPRESS_ClientesID");
```

-- Creating the EXPRESS_Economia table

```
CREATE TABLE "EXPRESS_Economia" (
    "EXPRESS_EconomiaID" INTEGER PRIMARY KEY ,
    "Express_Version" VARCHAR(15) NOT NULL ,
    "Cantidad" REAL NOT NULL ,
    "Ingreso" BIT NOT NULL ,
    "Area" INTEGER NOT NULL ,
    "Motivo" VARCHAR(200) NOT NULL ,
    "Fecha" DATE NOT NULL ,
    "Usuario" INTEGER NOT NULL );
```

```
CREATE INDEX "WDIDX_EXPRESS_Economia_Area" ON "EXPRESS_Economia"
("Area");
```

```
CREATE INDEX "WDIDX_EXPRESS_Economia_Usuario" ON "EXPRESS_Economia"
("Usuario");
```

-- Creating the EXPRESS_Material table

```
CREATE TABLE "EXPRESS_Material" (
    "EXPRESS_MaterialID" INTEGER PRIMARY KEY ,
```



```
"Express_Version" VARCHAR(15) NOT NULL ,
"Nombre" VARCHAR(100) NOT NULL UNIQUE );

-- Creating the EXPRESS_MaterialArea table
CREATE TABLE "EXPRESS_MaterialArea" (
    "EXPRESS_MaterialAreaID" INTEGER PRIMARY KEY ,
    "Express_Version" VARCHAR(15) NOT NULL ,
    "IdMaterial" INTEGER NOT NULL ,
    "IdArea" INTEGER NOT NULL ,
    "Unidades" INTEGER NOT NULL ,
    "NombreMaterial" VARCHAR(100) NOT NULL );
CREATE INDEX "WDIDX_EXPRESS_MaterialArea_IdMaterial" ON
"EXPRESS_MaterialArea" ("IdMaterial");
CREATE INDEX "WDIDX_EXPRESS_MaterialArea_IdArea" ON
"EXPRESS_MaterialArea" ("IdArea");

-- Creating the EXPRESS_NivelesTrabajadores table
CREATE TABLE "EXPRESS_NivelesTrabajadores" (
    "EXPRESS_NivelesTrabajadoresID" INTEGER PRIMARY KEY ,
    "Express_Version" VARCHAR(15) NOT NULL ,
    "Nombre" VARCHAR(100) NOT NULL UNIQUE );

-- Creating the EXPRESS_Pistas table
CREATE TABLE "EXPRESS_Pistas" (
    "EXPRESS_PistasID" INTEGER PRIMARY KEY ,
    "Express_Version" VARCHAR(15) NOT NULL ,
    "Nombre" VARCHAR(100) NOT NULL ,
    "Area" INTEGER NOT NULL ,
    "Estado" BIT NOT NULL );
CREATE INDEX "WDIDX_EXPRESS_Pistas_Area" ON "EXPRESS_Pistas" ("Area");
```

-- Creating the EXPRESS_Reservas table

```
CREATE TABLE "EXPRESS_Reservas" (  
    "EXPRESS_ReservasID" INTEGER PRIMARY KEY,  
    "Express_Version" VARCHAR(15) NOT NULL,  
    "Cliente" VARCHAR(9) NOT NULL,  
    "Pista" INTEGER NOT NULL,  
    "Fecha" DATE NOT NULL,  
    "Pagado" BIT NOT NULL,  
    "Hora" TIME NOT NULL );  
  
CREATE INDEX "WDIDX_EXPRESS_Reservas_Cliente" ON "EXPRESS_Reservas"  
("Cliente");  
  
CREATE INDEX "WDIDX_EXPRESS_Reservas_Pista" ON "EXPRESS_Reservas"  
("Pista");
```

-- Creating the EXPRESS_Trabajadores table

```
CREATE TABLE "EXPRESS_Trabajadores" (  
    "TrabajadoresID" INTEGER PRIMARY KEY,  
    "Nombre" VARCHAR(100) NOT NULL,  
    "Dni" VARCHAR(9) NOT NULL UNIQUE,  
    "Apellido1" VARCHAR(100) NOT NULL,  
    "Express_Version" VARCHAR(15) NOT NULL,  
    "Apellido2" VARCHAR(100) NOT NULL,  
    "Nacimiento" DATE NOT NULL,  
    "Area" INTEGER NOT NULL,  
    "Nivel" INTEGER NOT NULL,  
    "Direccion" VARCHAR(200) NOT NULL,  
    "NSS" VARCHAR(50) NOT NULL UNIQUE,  
    "EntradaEmpresa" DATE NOT NULL,  
    "Password" VARCHAR(50) NOT NULL );
```

```
CREATE INDEX "WDIDX_EXPRESS_Trabajadores_Area" ON "EXPRESS_Trabajadores"
("Area");
```

```
CREATE INDEX "WDIDX_EXPRESS_Trabajadores_Nivel" ON "EXPRESS_Trabajadores"
("Nivel");
```

-- Creating the EXPRESS_Trabajadores_EXPRESS_Areas table

```
CREATE TABLE "EXPRESS_Trabajadores_EXPRESS_Areas" (
```

```
    "Express_Version" VARCHAR(15) NOT NULL ,
```

```
    "EXPRESS_AreasID" INTEGER NOT NULL ,
```

```
    "TrabajadoresID" INTEGER NOT NULL );
```

```
CREATE INDEX "WDIDX_EXPRESS_Trabajadores_EXPRESS_Areas_EXPRESS_AreasID"
ON "EXPRESS_Trabajadores_EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
CREATE INDEX "WDIDX_EXPRESS_Trabajadores_EXPRESS_Areas_TrabajadoresID"
ON "EXPRESS_Trabajadores_EXPRESS_Areas" ("TrabajadoresID");
```

-- Creating the HisotiralReservas table

```
CREATE TABLE "HisotiralReservas" (
```

```
    "HisotiralReservasID" INTEGER PRIMARY KEY ,
```

```
    "idCliente" INTEGER NOT NULL ,
```

```
    "Fecha" DATE NOT NULL ,
```

```
    "Hora" TIME NOT NULL ,
```

```
    "Pista" INTEGER NOT NULL ,
```

```
    "Motivo" VARCHAR(200) NOT NULL );
```

```
CREATE INDEX "WDIDX_HisotiralReservas_idCliente" ON "HisotiralReservas"
("idCliente");
```

```
CREATE INDEX "WDIDX_HisotiralReservas_Pista" ON "HisotiralReservas" ("Pista");
```

-- Creating the HistoricoClientes table

```
CREATE TABLE "HistoricoClientes" (
```

```
    "HistoricoClientesID" INTEGER PRIMARY KEY ,
```

```
    "Nombre" VARCHAR(100) NOT NULL ,
```

```
"Apellido1" VARCHAR(100) NOT NULL ,  
"Apellido2" VARCHAR(100) NOT NULL ,  
"Nacimiento" DATE NOT NULL ,  
"Direccion" VARCHAR(200) NOT NULL ,  
"Telefono" NUMERIC(10,0) NOT NULL ,  
"Dni" VARCHAR(9) NOT NULL UNIQUE );
```

-- Creating the Horarios table

```
CREATE TABLE "Horarios" (  
    "HorariosID" INTEGER PRIMARY KEY ,  
    "IdTrabajador" INTEGER NOT NULL ,  
    "Fecha" DATE NOT NULL ,  
    "HoraInicial" TIME NOT NULL );  
CREATE INDEX "WDIDX_Horarios_IdTrabajador" ON "Horarios" ("IdTrabajador");
```

-- Creating the Pedidos table

```
CREATE TABLE "Pedidos" (  
    "PedidosID" INTEGER PRIMARY KEY ,  
    "NPedido" INTEGER NOT NULL ,  
    "IdMaterial" INTEGER NOT NULL ,  
    "Cantidad" INTEGER NOT NULL ,  
    "IdTrabajador" INTEGER NOT NULL ,  
    "Confirmado" BIT NOT NULL ,  
    "Material" VARCHAR(100) NOT NULL ,  
    "IdArea" INTEGER NOT NULL );  
CREATE INDEX "WDIDX_Pedidos_NPedido" ON "Pedidos" ("NPedido");  
CREATE INDEX "WDIDX_Pedidos_IdMaterial" ON "Pedidos" ("IdMaterial");  
CREATE INDEX "WDIDX_Pedidos_IdTrabajador" ON "Pedidos" ("IdTrabajador");  
CREATE INDEX "WDIDX_Pedidos_IdArea" ON "Pedidos" ("IdArea");
```

--Integrity Constraints

```
ALTER TABLE "HisotiralReservas" ADD FOREIGN KEY ("idCliente") REFERENCES  
"EXPRESS_Clientes" ("EXPRESS_ClientesID");
```

```
ALTER TABLE "HisotiralReservas" ADD FOREIGN KEY ("Pista") REFERENCES  
"EXPRESS_Pistas" ("EXPRESS_PistasID");
```

```
ALTER TABLE "EXPRESS_Reservas" ADD FOREIGN KEY ("Pista") REFERENCES  
"EXPRESS_Pistas" ("EXPRESS_PistasID");
```

```
ALTER TABLE "EXPRESS_Pistas" ADD FOREIGN KEY ("Area") REFERENCES  
"EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
ALTER TABLE "EXPRESS_Economia" ADD FOREIGN KEY ("Area") REFERENCES  
"EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
ALTER TABLE "EXPRESS_Economia" ADD FOREIGN KEY ("Usuario") REFERENCES  
"EXPRESS_Trabajadores" ("TrabajadoresID");
```

```
ALTER TABLE "EXPRESS_MaterialArea" ADD FOREIGN KEY ("IdArea") REFERENCES  
"EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
ALTER TABLE "EXPRESS_MaterialArea" ADD FOREIGN KEY ("IdMaterial")  
REFERENCES "EXPRESS_Material" ("EXPRESS_MaterialID");
```

```
ALTER TABLE "EXPRESS_Clientes_EXPRESS_Areas" ADD FOREIGN KEY  
("EXPRESS_AreasID") REFERENCES "EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
ALTER TABLE "EXPRESS_Clientes_EXPRESS_Areas" ADD FOREIGN KEY  
("EXPRESS_ClientesID") REFERENCES "EXPRESS_Clientes" ("EXPRESS_ClientesID");
```

```
ALTER TABLE "EXPRESS_Trabajadores_EXPRESS_Areas" ADD FOREIGN KEY  
("EXPRESS_AreasID") REFERENCES "EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
ALTER TABLE "EXPRESS_Trabajadores_EXPRESS_Areas" ADD FOREIGN KEY  
("TrabajadoresID") REFERENCES "EXPRESS_Trabajadores" ("TrabajadoresID");
```

```
ALTER TABLE "EXPRESS_Trabajadores" ADD FOREIGN KEY ("Nivel") REFERENCES  
"EXPRESS_NivelesTrabajadores" ("EXPRESS_NivelesTrabajadoresID");
```

```
ALTER TABLE "EXPRESS_Clientes" ADD FOREIGN KEY ("Monitor") REFERENCES  
"EXPRESS_Trabajadores" ("TrabajadoresID");
```

```
ALTER TABLE "EXPRESS_Reservas" ADD FOREIGN KEY ("Cliente") REFERENCES  
"EXPRESS_Clientes" ("Dni");
```

```
ALTER TABLE "Pedidos" ADD FOREIGN KEY ("IdArea") REFERENCES  
"EXPRESS_Areas" ("EXPRESS_AreasID");
```

```
ALTER TABLE "Horarios" ADD FOREIGN KEY ("IdTrabajador") REFERENCES  
"EXPRESS_Trabajadores" ("TrabajadoresID");
```

```
ALTER TABLE "Pedidos" ADD FOREIGN KEY ("IdTrabajador") REFERENCES  
"EXPRESS_Trabajadores" ("TrabajadoresID");
```

```
ALTER TABLE "Pedidos" ADD FOREIGN KEY ("IdMaterial") REFERENCES  
"EXPRESS_Material" ("EXPRESS_MaterialID");
```

```
ALTER TABLE "ActividadesOfertadas" ADD FOREIGN KEY ("idTrabajador")  
REFERENCES "EXPRESS_Trabajadores" ("TrabajadoresID");
```

```
ALTER TABLE "ActividadesOfertadas" ADD FOREIGN KEY ("idPista") REFERENCES  
"EXPRESS_Pistas" ("EXPRESS_PistasID");
```

```
ALTER TABLE "ActividadesOfertadas_EXPRESS_Clientes" ADD FOREIGN KEY  
("EXPRESS_ClientesID") REFERENCES "EXPRESS_Clientes" ("EXPRESS_ClientesID");
```

```
ALTER TABLE "ActividadesOfertadas_EXPRESS_Clientes" ADD FOREIGN KEY  
("ActividadesOfertadasID") REFERENCES "ActividadesOfertadas"  
("ActividadesOfertadasID");
```

```
ALTER TABLE "ActividadesOfertadas" ADD FOREIGN KEY ("idActividad")  
REFERENCES "Actividades" ("ActividadesID");
```

7. Object oriented model

7.1 Use case diagram

7.1.1 Client diagram

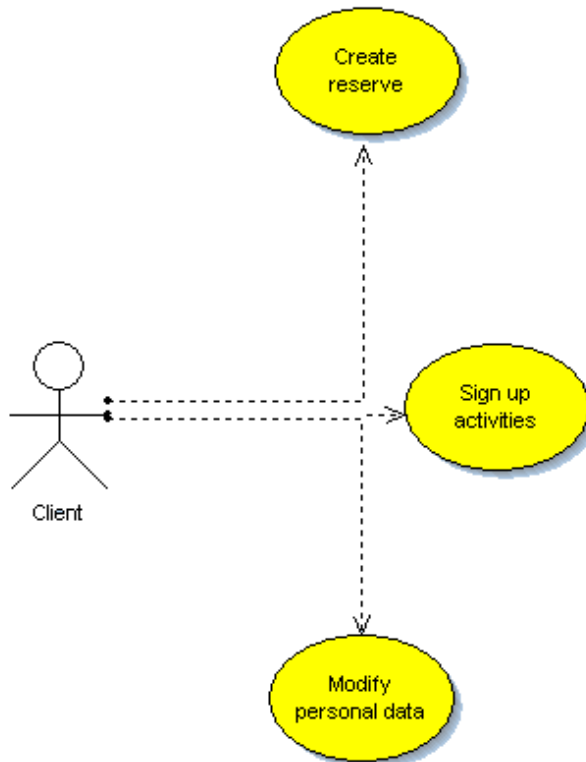


Figure 15 – Client user diagram

Use Case Name :	Create Reserve
Actors :	Client, Employer
Description :	This use case describes the process of a make a new reserve by a client. The client can choose an available area and choose a time, available also, for make his reserve. After that, client should pay the reserve and that generate an entry. If client wants to cancel the reserve, its possible but he loose 25% of price of the reserve.

Use Case Name :	Sign up activities
Actors :	Client
Description :	This use case describes the process of sign up in a activity by a client. The sport center offers a list of available activities. Client can choose all that he wants. If client wants to cancel a reserve, its possible. Same way he sign up there is an option for cancel this enrolment

Use Case Name :	Modify Personal data
Actors :	Client
Description :	This use case describes the process of the possibility of change some personal data like password or address. Client, alter login into system, can change their personal data.

7.1.2 Employer diagram

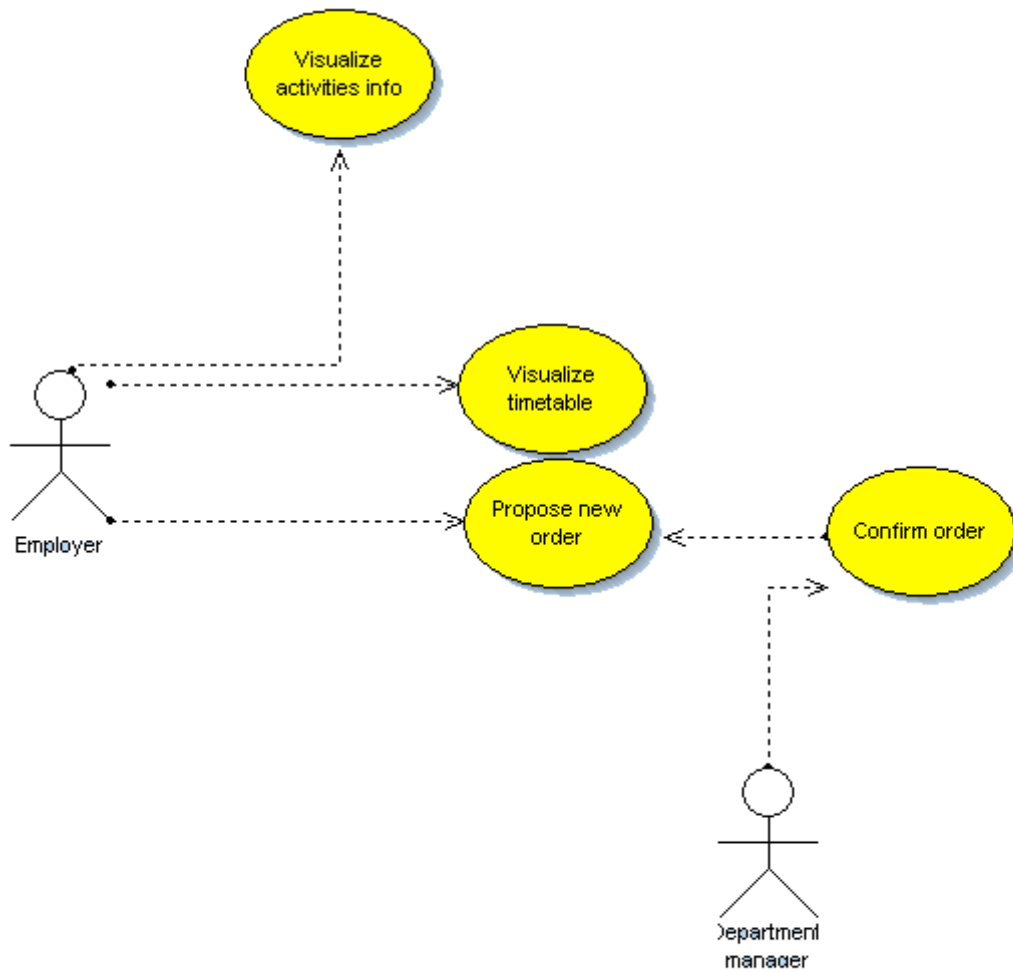


Figure 16 – employer user diagram

Use Case Name :	Visualize Activities info
Actors :	Employer, Department manager
Description :	This use case describes the process of show information about activities which employer it's the responsible. For example, he can look the timetable for these activities.

Use Case Name :	Visualize timetable
Actors :	Employer
Description :	This use case describes the process which an employer can look his timetable for the actual week

Use Case Name :	Propose new order
Actors :	Employer
Description :	This use case describes the process of order new equipment. He can add whatever ítems that he needs, and after that he should put prices for ítems which are requested. This order will have a status called "pending" until department manager confirm it.

7.1.3 Department Manager diagram

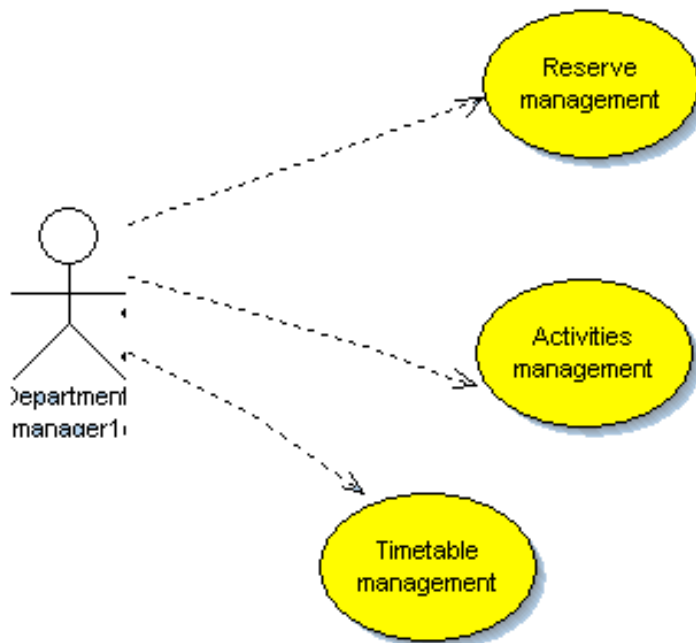


Figure 17– department manager user diagram

Use Case Name :	Confirm new order
Actors :	Department Manager
Description :	This use case describes the process of confirm a “pending” order. Application show in red color which orders are pending. Then the manager can confirmed the order, and immediatelly occurs a expenditure for value of the order.

Use Case Name :	Reserve Management
Actors :	Department manager
Description :	This use case describes the process of reserve management. Department manager can cancel any reserve produced in the system. This is so useful because if any activity needs a area and it’s reserved, he can cancel it and alter make a new reserve.

Use Case Name :	Activities Management
Actors :	Department Manager
Description :	This use case describes the process of activities management. First feature is define new activities. Second one is make avaiable the activities for give the possibilty to users to sign up. In addition he can cancel this availability and delete activities.

Use Case Name :	Timetable Management
Actors :	Department Manager
Description :	This use case describes the process of timetable Management. Principal feature is that he can make timetable for his employers.

7.1.4 Administration diagram

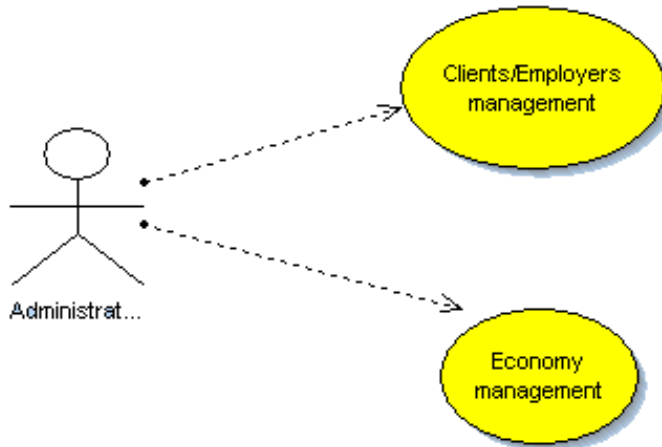
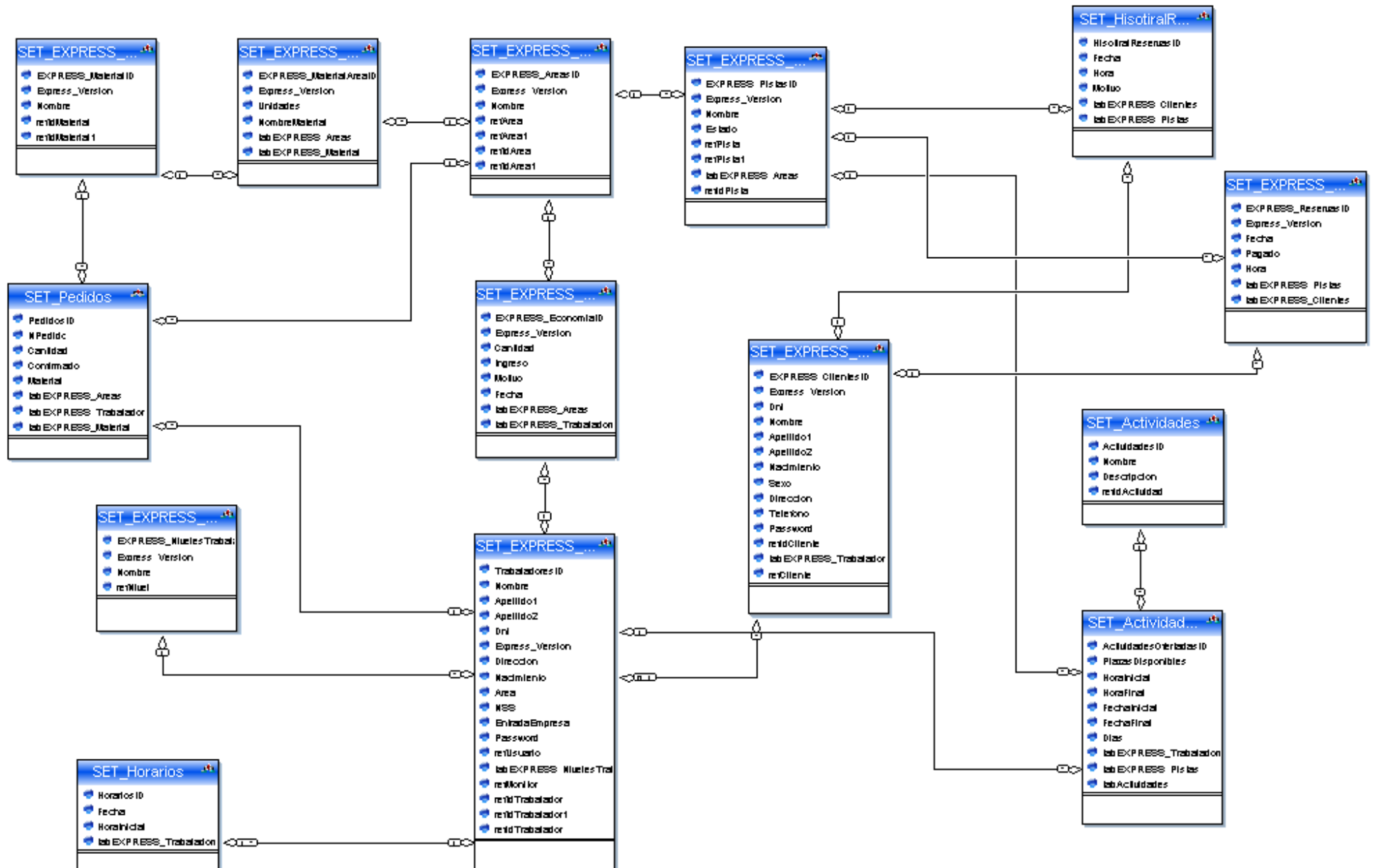


Figure 18 - administration user diagram

Use Case Name :	Clients/employers Management
Actors :	Administration
Description :	This use case describes the process of personal Management. Administration can add new employers/users, delete them and modify their data. Furthermore, administration can list clients/employers.

Use Case Name :	Economy management
Actors :	Administration
Description :	This use case describes the process of economy Management. Principal feature is add a new entry/expenditure and it's possible show a log of entries/expenditures for control.

7.2 Class Diagram



7.3. Sequence diagrams for some use cases

7.3.1. New reserve diagram

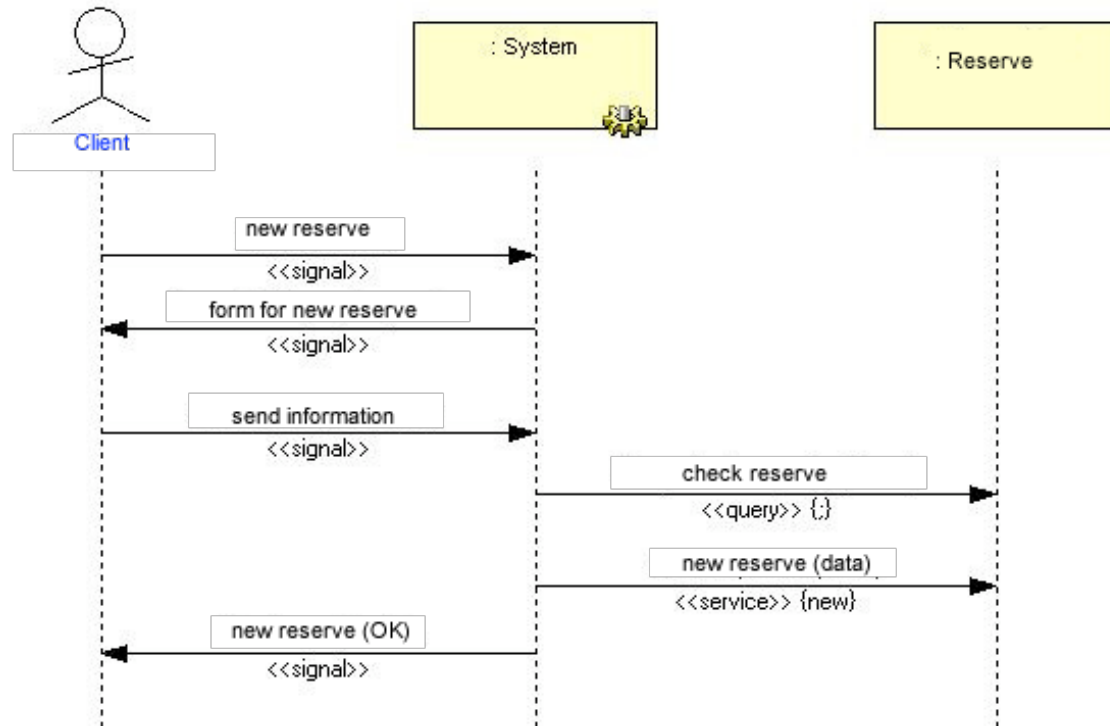


Figure 19 – new reserve sequence diagram

7.3.2. Sign up activity diagram

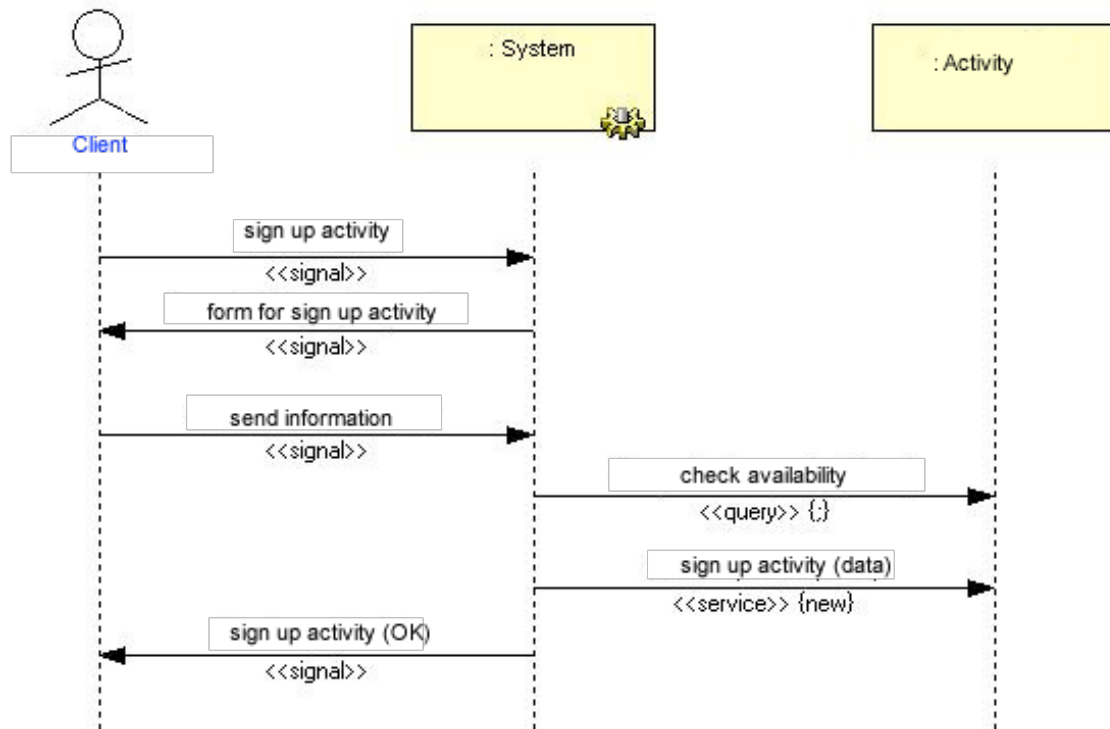


Figure 20 – sign up activity sequence diagram

7.3.3. List clients diagram

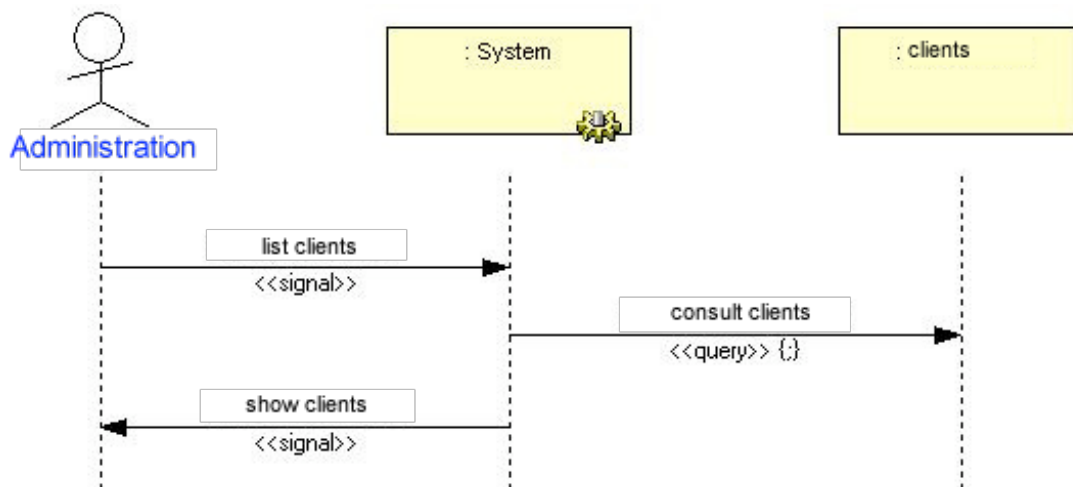


Figure 21 – list clients sequence diagram

7.3.4. Hire new employer diagram

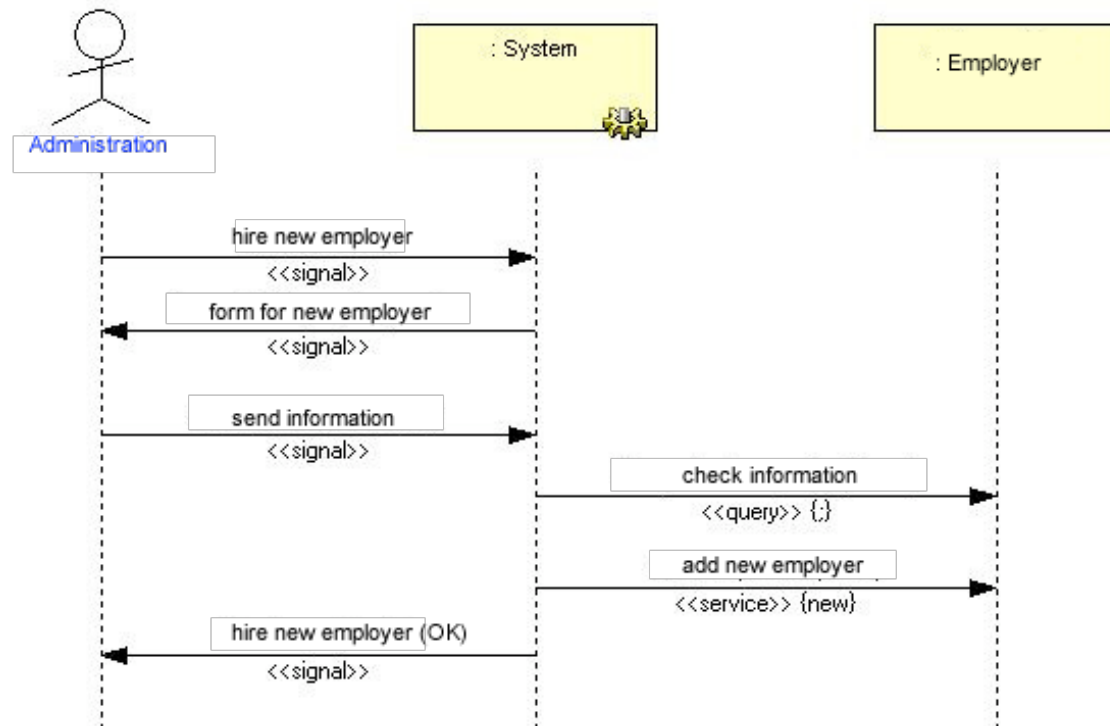


Figure 22 – hire new employer sequence diagram

7.4 Class declarations code generated from the class diagram.

```

STSET_Actividades is a structure
    ActividadesID is int //Identifier of Actividades
    Nombre is string //Nombre de la actividad
    Descripcion is string //Descripcion de la actividad
    refIdActividad is dynamic array of 0 STSET_ActividadesOfertadas
dynamic
END

```

```

STSET_ActividadesOfertadas is a structure
    ActividadesOfertadasID is int //Identifier of
ActividadesOfertadas
    PlazasDisponibles is int //Indica el numero de plazas que aun
dispone esta actividad
    HoraInicial is Time //Horainicial
    HoraFinal is Time //Horafinal
    FechaInicial is Date //Fecha en la que inicia la actividad
    FechaFinal is Date //Fecha cuando la actividad terminará
    Dias is string //Dias en los que se realiza la actividad
    tabEXPRESS_Trabajadores is STSET_EXPRESS_Trabajadores dynamic
    tabEXPRESS_Pistas is STSET_EXPRESS_Pistas dynamic
    tabActividades is STSET_Actividades dynamic
END

```

```

STSET_EXPRESS_Areas is a structure
    EXPRESS_AreasID is int //Identifier of EXPRESS_Areas
    Express_Version is string //Express_Version
    Nombre is string //Nombre del Area del complejo deportivo
    refArea is dynamic array of 0 STSET_EXPRESS_Pistas dynamic
    refArea1 is dynamic array of 0 STSET_EXPRESS_Economia dynamic
    refIdArea is dynamic array of 0 STSET_EXPRESS_MaterialArea
dynamic
    refIdArea1 is dynamic array of 0 STSET_Pedidos dynamic
END

```

```

STSET_EXPRESS_Clientes is a structure
    EXPRESS_ClientesID is int //Identifier of EXPRESS_Clientes
    Express_Version is string //Express_Version
    Dni is string //Dni
    Nombre is string //Express_Nombre
    Apellido1 is string //Apellido1
    Apellido2 is string //Apellido2
    Nacimiento is Date //Nacimiento
    Sexo is string //Sexo
    Direccion is string //Direccion
    Telefono is 4-byte unsigned int //Telefono
    Password is string //Contraseña con la que el usuario entra al
programa
    tabEXPRESS_Trabajadores is STSET_EXPRESS_Trabajadores dynamic
    refCliente is dynamic array of 0 STSET_EXPRESS_Reservas dynamic
END

```

```

STSET_EXPRESS_Economia is a structure
    EXPRESS_EconomiaID is int //Identifier of EXPRESS_Economia
    Express_Version is string //Express_Version
    Cantidad is 4-byte real //Cantidad de gasto o ingreso

```



```

    Ingreso is boolean //Ingreso (si esta chequeado es que es ingreso
sino es gasto)
    Motivo is string //Motivo del gasto o del ingreso
    Fecha is Date //Fecha del ingreso o desembolso
    tabEXPRESS_Areas is STSET_EXPRESS_Areas dynamic
    tabEXPRESS_Trabajadores is STSET_EXPRESS_Trabajadores dynamic
END

```

```

STSET_EXPRESS_Material is a structure
    EXPRESS_MaterialID is int //Identifier of EXPRESS_Material
    Express_Version is string //Express_Version
    Nombre is string //Nombre del utensilio
    refIdMaterial is dynamic array of 0 STSET_EXPRESS_MaterialArea
dynamic
    refIdMaterial1 is dynamic array of 0 STSET_Pedidos dynamic
END

```

```

STSET_EXPRESS_MaterialArea is a structure
    EXPRESS_MaterialAreaID is int //Identifier of
EXPRESS_MaterialArea
    Express_Version is string //Express_Version
    Unidades is int //Unidades
    NombreMaterial is string //Nombrematerial
    tabEXPRESS_Areas is STSET_EXPRESS_Areas dynamic
    tabEXPRESS_Material is STSET_EXPRESS_Material dynamic
END

```

```

STSET_EXPRESS_Pistas is a structure
    EXPRESS_PistasID is int //Identifier of EXPRESS_Pistas
    Express_Version is string //Express_Version
    Nombre is string //Nombre del utensilio
    Estado is boolean //Estado (Si esta chequeado indica que la pista
esta disponible para el uso)
    refPista is dynamic array of 0 STSET_EXPRESS_Reservas dynamic
    tabEXPRESS_Areas is STSET_EXPRESS_Areas dynamic
    refidPista is dynamic array of 0 STSET_ActividadesOfertadas
dynamic
END

```

```

STSET_EXPRESS_Reservas is a structure
    EXPRESS_ReservasID is int //Identifier of EXPRESS_Reservas
    Express_Version is string //Express_Version
    Fecha is Date //Fecha de la reserva
    Pagado is boolean //Pagado (si el check esta habilitado,
significa que si que se ha pagado)
    Hora is Time //Hora de la reserva
    tabEXPRESS_Pistas is STSET_EXPRESS_Pistas dynamic
    tabEXPRESS_Clientes is STSET_EXPRESS_Clientes dynamic
END

```

```

STSET_EXPRESS_Trabajadores is a structure
    TrabajadoresID is int //Identifier of EXPRESS_Trabajadores
    Nombre is string //Nombre del trabajador
    Apellido1 is string //Primer apellido del trabajador
    Apellido2 is string //Segundo apellido del trabajador
    Dni is string //Dni
    Express_Version is string //Express_Version
    Dirección is string //Direccion del trabajador
    Nacimiento is Date //Fecha de Nacimiento
    Area is int //Area donde trabaja este empleado
    NSS is string //Numero de la Seguridad Social
    EntradaEmpresa is Date //Fecha en la que el trabajador entró en
la empresa
    Password is string //Contraseña con la que el usuario entra al
programa

```

```
    refUsuario is dynamic array of 0 STSET_EXPRESS_Economia dynamic
    refMonitor is dynamic array of 0 STSET_EXPRESS_Clientes dynamic
    refIdTrabajador is dynamic array of 1 STSET_Horarios dynamic
    refIdTrabajador1 is dynamic array of 0 STSET_Pedidos dynamic
    refidTrabajador is dynamic array of 0 STSET_ActividadesOfertadas
dynamic
END

STSET_Horarios is a structure
    HorariosID is int //Identifier of Horarios
    Fecha is Date //Dia del horario
    HoraInicial is Time //Hora en la que entra a trabajar
    tabEXPRESS_Trabajadores is STSET_EXPRESS_Trabajadores dynamic
END

STSET_Pedidos is a structure
    PedidosID is int //Identifier of Pedidos
    NPedido is int //NPedido
    Cantidad is int //Cantidad de unidades a pedir del material
    Confirmado is boolean //Confirmado (si esta checkeado es que esta
confirmado)
    Material is string //Nombre del material (servira en caso de que
el usuario eliga la opicion de otros)
    tabEXPRESS_Areas is STSET_EXPRESS_Areas dynamic
    tabEXPRESS_Trabajadores is STSET_EXPRESS_Trabajadores dynamic
    tabEXPRESS_Material is STSET_EXPRESS_Material dynamic
END
```

