9. Developing a Life Cycle Document Management System for Construction

9.1. Introduction

This chapter deals with the description of the *Life cycle Document Management System for construction*. Notwithstanding, the goal here is to define the System Requirements, i.e. the elements, functionalities, and objectives of the database, the structure of the database based on the *Concept Model for Information Flow* (Chapter 7), and to set the relations within this information. Once the database is defined, the aim is to implement it into a web based platform for to allow all the interested parties Internet access to the System.

9.2. An overview to Database Management Systems (DBMS)

9.2.1. Introduction to databases

A database is a collection of related information. A database system basically is just a computerized record-keeping system. The database itself can be regarded as a kind of electronic filing cabinet, i.e., a repository or container for a collection of computerized data files. Users of the system can perform several operations on such files, for example:

- Adding new, empty files to the database
- Inserting data into existing files
- Retrieving data from existing files
- Changing data in existing files
- Deleting data in existing files
- Removing existing files from the database.

A database system involves four major components: data, hardware, software and users. (Date 2000).

- **Data**

Database systems are available on machines that range from the smallest personal computers to the largest mainframes. The facilities provided by any given system are to some extent determined by the size and power of the underlying machine. In particular, systems on larger machines tend to be multi-users, whereas those on smaller machines tend to be single-user. A single-user system is one in which at most one user can access the database at any given time; a multi-user system is one in which many users can access the database at the same time. In general, data in the database are integrated and shared. By integration, it's meant that the database can be thought of as a unification of several otherwise distinct files, any redundancy among those files having been at least partly eliminated. By shared, it's meant that individual pieces of data in the database can be shared among different users, in the sense that each user can have access to the same piece of data for different purposes.

- **Hardware**

The hardware components of the system consist of the secondary storage volumes that are used to hold the stored data, together with the associated I/O devices (disk drives, etc.). The programs themselves can be conventional batch applications, or they can be online applications, whose purpose is to allow an end user to access the database from an online workstation or terminal. Most modern applications belong to the online variety.

- **Software**

Between the physical database itself and the users of the system there is a layer of software, known as database manager or database server or, most commonly, database management system (DBMS). Every request for access to the database is handled by the DBMS, the most important software component in the overall system.

- **Users**

There are three broad classes of users:

First, there are the application programmers, responsible for writing database application programs in some programming language such as COBOL, C++, and Java. Such programs access the database by issuing the appropriate request –typically, an SQL statement- to the DBMS. The programs themselves can be conventional batch applications, or they can be online applications, whose purpose is to allow an end user to access the database from an online workstation or terminal.

The second class is the end users, who interact with the system from online workstations or terminals. An end user can access the database via one of the aforementioned online applications, or can use an interface provided as an integral part of the database system

software. Such vendor-provided interfaces are also supported by means of online applications, but those applications are built-in, not user-written. Most database systems include at least one such built-in application, namely a query language processor, by which the user can interactively issue database requests such as SELECT and INSERT to the DBMS. SQL language is a typical example of a database query language.

The third class of user is the database administrator (DBA). A DBA directs or performs all the activities related to maintaining a successful database environment. Responsibilities include designing, implementing, and maintaining the database system; establishing policies and procedures pertaining to the management, security, maintenance, and use of the database management system; and training employees in database management and use.

9.2.2. Relational DBMS

DBMS organize and structure data so as to be used and manipulated by users and application programmes. DMBS data structure and access technique is called Data Model. The DM of a DBMS defines its 'personality'. (Groff & Weinberg 1990).

When designing a data base the first stage is to choose the adequate Data Model, depending on the tasks to be executed by the database.

Database Systems (DS) can be conveniently categorized according to the data structures and operators they present to the user. According to this scheme, the oldest systems fall into three broad categories: relational, hierarchic, and network.

In a relational system the user views the data as tables, and views nothing but tables. By contrast, in a non-relational system the user views other data structures, either instead of, or in addition to, the tables of a relational system. Those other structures, in turn, require another operator to manipulate them. For example, in a hierarchic system the data is represented to the user in the form of a set of tree structures (hierarchies). In a hierarchical system there are relations for each data which is stored in individual registers. Therefore, when consulting, registers should be searched one by one, and when there is a great deal of data and relations between them this system becomes slow and inefficient. We do not discuss these categories in detail because, at least from a technological point of view, they must be regarded as obsolete. However, a schema of data organization is presented in order to show the differences.

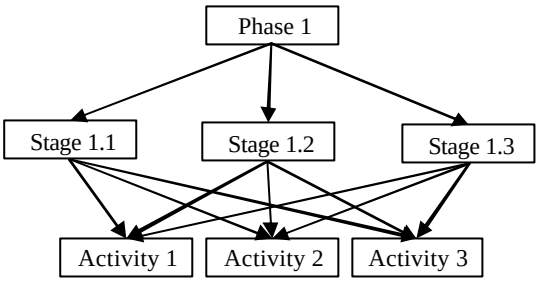


Figure 25. Example of a hierarchical structure

table PHASE	table STAGE	table ACTIVITY
Name phase: Description:	Name stage: Description:	Name activity: Description:

Figure 26. Example of relational organization of tables

The first relational products appeared in the late 1970s and early 1980s. Currently, most database systems are relational and run on just about every kind of hardware and software platform available.

Advances in database technology and processing offer opportunities for using information in a flexible and efficient way when data is organized and stored in relational structures. A relational data model defines what the data is rather than how it is used, because data is used in multiple applications to serve multiple functions.

9.2.3. Steps to develop a DBMS

The overall approach to the semantic modeling problem can be characterized in terms of the following four steps:

- First, we attempt to identify the necessities and requirements of the database.

- Then, we should **identify a set of semantic concepts** that seem to be useful in talking informally about the real world. It is called **Conceptual Design** and aims to obtain a structure of the information of the future DB. In this step the terms: entities, attributes and relations, are used. **Entities** are objects belonging to the real world that can be distinguished from the rest of objects. These entities can usefully be classified into **entity types**. The advantage of

such classification is that all the entities of a given type will have certain properties or **attributes** in common. Every entity has a special an identifier property, i.e., every entity has an **identity**. Any entity can be related to other entities by means of **relationships**.

- Next, we should **devise a set of corresponding symbolic objects** that can be used to represent the foregoing semantic concepts.

- Then, we should **devise a set of formal, general integrity rules,** to go along with those formal objects.

- Finally, we should **develop a set of formal operators** for manipulating those formal objects.

The best-known semantic modeling approach and the most widely used is the **entity/relationship (E/R)** approach, based on the E/R model introduced by Chen in 1976. The E/R model introduces the concept of E/R diagram, which constitutes a technique for representing the logical structure of a database in a pictorial manner. The E/R diagramming technique is explained below:

- **Entity:** Each entity type is shown as a rectangle containing the name of the entity type in question. Each entity must have a candidate key to identify it which is called **primary key**. Elements are the discrete pieces of data which describe and define entities. An attribute is an intrinsic characteristic of an entity. Elements define the attributes of entities.

- **Properties:** Properties are shown as ellipses containing the name of the property in question and are attached to the relevant entity or relationship by means of a solid line. The ellipse border is dotted or dashed if the property is derived, and doubled if it is multi-valued. If the property is composite, its component properties are shown as further ellipses, connected to the ellipse for the composite property in question by means of further solid lines. Key properties are underlined. The value sets corresponding to properties are not shown.

- **Relationships:** Each relationship type is shown as a diamond containing the name of the relationship type in question. The diamond border is doubled if the relationship in question is between a weak entity type and the entity type on which its existence depends. The participants in each relationship are connected into the relevant relationship by means of solid lines; each such line is labelled '1' or 'M' to indicate whether the relationship is one-to-one, many-to-one, etc. The line is doubled if the participation is total. In the one-to-one relationship only one element of the first entity can have a relation to only one element of the second entity. In the many-to-one relationship there are many different elements of the first entity that have some kind of relationship to only one element of the second entity.

Afterwards, when defining the symbolic objects, all the elements of the conceptual concept turn into tables.

To define correctly a relation database it should obey a group of fundamental premises so we should **devise a set of corresponding symbolic objects.**
It's important to define accurately all the elements, tables, primary keys, foreign keys and relations of all the stored data.

A **table** is a grouping of related data organized in fields (columns) and records (rows) on a datasheet. By using a common field in two tables the data can be combined. Many tables can be stored in a single database. A table is the ground for organizing a relational database, a bidimensional disposition of rows and columns of the data. Each entity will be represented as a table. Each table of a relational database should have a unique name that identifies its content. Each row represents a unique entity. Each column represents a set of data from the same type. To identify an entity we should create the **primary key,** which is an attribute of the entity and describes the uniqueness of each entity.

A **field** is a column on a datasheet and defines a data type for a set of values in a table. When creating the relationship between tables, a field from one table is related to another field of another table; the data that is linked between two tables is called **foreign key**. Each field should contain only one value (atomicity) and all the fields should depend only from the primary key.

A **record** in a row on a datasheet is a set of values defined by fields. Every record in a table must have a **primary key** that differentiates it from every other record in the table. In some cases it is only necessary to designate an existing field as the primary key, if you are certain that every record in the table will have a different value for that particular field.

To prevent the duplication of information in a database by repeating fields in more than one table, **table relationships** can be established to link fields of tables together.

9.3. Definition of the System Requirements

From the Conceptual Model for Information Flow described in Chapter 7, the elements and functionalities of the *Life cycle Document Management System for Construction* are defined. From Chapter 7 the following conclusions were obtained:

- The project document organization should be organized by its life cycle, which is divided into phases / stages.

- The type of information, and the area of the project where a piece of information belongs, should also be considered and is stated as activities / subactivities.

- The actors that take part in a construction project should be also defined.

- From each document, other information (metadata) to track, find, manage and use these data is also relevant. These metadata is divided basically into: Document Name, Description, Late submittal Date, Attribute and Type of document.

- Once the document is located and its main characteristics are defined, the relations of each actor with the document -which are described as 'responsibilities'- should be also taken into consideration. Actually, although there are a vast number of participants in a construction project they have been summarized into three: Client, Designer and Constructor, and are called actors. Each actor can have a different responsibility concerning the document; responsibilities can be: create and receive. This characteristic defines the responsibility for uploading a document in the EDMS and/or for delivering it. The actor responsible for the creation of a document must be aware of who will need the document and when.

- Moreover, there are different types of procurement arrangements in a construction project. So, depending on the contractual arrangements, different participants take part assuming different responsibilities: In the **Traditional Procurement arrangement** the client has a direct contractual relationship with most participants. In the **Turnkey Project arrangement**, the client delegates all the design and construction responsibilities to external consultants. In **Professional Construction Management arrangement** there is no main contractor interposed between the owner and the various specialist subcontractors. The construction manager becomes the principal consultant coordinating the entire procurement process.

Once the actors' responsibilities are defined, these actors should be split into the roles they are going to develop in the project. I.e., depending on the contractual arrangement, the client can perform different roles such as that of a construction manager, which can also be carried out by the designer or the constructor in another type of contractual arrangement.

On the whole, the scope of this system is to:

1. **Create a folder structure for whatever construction project,** to be used in the corporate server, in the individual PCs or in the Web Based Project Management System.

The aim is to have the same folder and file structure for whatever actor working in WPMS. This tool aims to improve the internal document management of the company and the interaction between each company (agent) and the WPMS. By choosing some inputs such as the type of contract and the actors that are going to partake in the project, the system generates a matrix where to place each document along the life cycle. To make it easier and in a more traditional way, this organization of information will be downloaded into the user's PC and it automatically creates the structure of archives, folders, subfolders and so on.

This structure will also be downloaded or incorporated into whatever Web based Project Management System. The main objective is to create the same folder structure in each user's PC and in the Web based Project Management System.

2. **Consult** whatever information related to the life cycle of the project, document, etc.

Users are able to define some inputs such as responsibility for the document (create, receive), project phase (inception, design, etc.), stage (general design, detailed design, etc.), activity (costs, risks, quality, etc.), subactivity (communication, documentation, etc.) and type of document (letter, drawing, etc.). Then the system returns the document organization for these inputs.

For example, a designer might want to know when must he/she deliver the specifications of the project; he/she filters, then, the data by type of project -in this case: specifications- and by the responsibility for the project -in this case: create-, and the system tells him/her in which stage of the life cycle, activity and subactivity to deliver the specifications.

Another example might be the case of a client wanting to know which documents will he/she receive concerning a specific activity, e.g. costs. The systems returns all the Stages where the client will receive anything related to costs.

9.4. Database Design Criteria

Defining the process life cycle as Phases / Stages doesn't mean that for starting a phase the previous one should be finished. It only gives a temporal character. The same happens with the Stages. It is possible to have overlapping between Phases or Stages.

All the information to be represented in this database is like an index. If we represent it as a tree diagram we obtain the following organization of information:

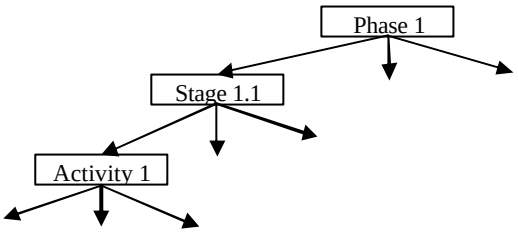


Figure 27. Tree diagram of a possible alternative

Then, the relationships between entities would be 1:N, which means that from the element of the first entity exists N relations that derive from the first one to N registers.

This structure can be translated to the following index.
1 Phase 1
1.1 Stage 1
1.1.1 Activity 1
1.1.1.1 Subactivity 1
1.1.1.1.1 Document 1

A relational database works with bidimensional tables but this information can be represented as a matricial structure.

Actually, activities and subactivities are repeated in each phase and stage, and the same happens with documents. We can have the document 'Minutes' in different stages like in 'Conceptual Definition' or in 'Construction', and these documents can be stored in different activities like 'Contractings' and 'Quality', and in different subactivities such as 'Communication' and 'Organization'.

This doesn't mean that there are redundancies of activities in different phases, but only that the entities names such as activity, subactivity, and documents, are repeated in different Phases and Stages but not their contents.

Table 17. Examples of repetition of different elements from different tables

Phases	Stage	Activity	Subactivity	Document
Execution	Construction	Quality	Organization	*Minutes*
Conception	Conceptual definition	Contractings	Communication	*Minutes*

Phases	Stage	Activity	Subactivity	Document
Execution	*Construction delivery*	*Costs*	Documentation	Letters
Execution	*Construction delivery*	*Costs*	Communication	Turnover

Consequently, the relation between Phases / Stages and Activities / Subactivities adopts the following form:

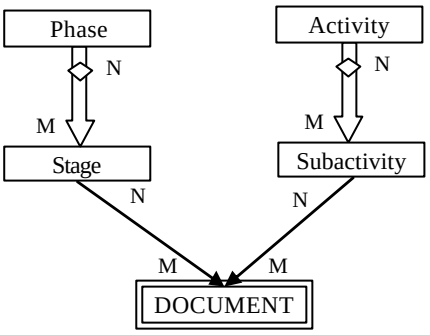


Figure 28. E/R diagram to organize documents along the life cycle

Once the document is defined and placed in the life cycle of the construction Project, we should analyze the responsibilities each participant has in relation to the document. As stated in Chapter 7, although there are many participants in a construction project we have reduced them into three: Client, Designer and Constructor, and are called actors. Each actor can have a different responsibility for the document, so we have divided the responsibilities into two: create and receive.

Another condition of the responsibility of each actor is the contractual arrangement of the project. Depending on the contract, each actor will partake in the project assuming different responsibilities.

This relation between Contracts, Agents, and Documents, is summarized in the following entity/relationship diagram:

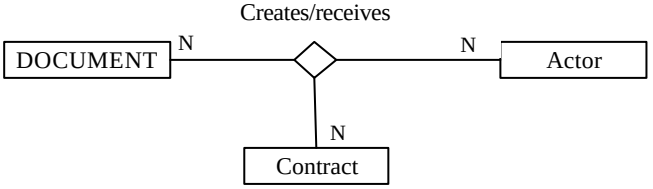


Figure 29. E/R diagram between Documents, Contracts, and Agents

From each document we also want to know which document and type of documents does it relate to.

Table 18. Examples of related documents

Phases	Stage	Activity	Subactivity	Document	Depends of
Conception	Feasibility Study	Project	Documentation	Annex	*Memory*
Execution	Construction	Advance	Monitoring and Control	Defects list	*Non conformities*

The entity / relationship diagram to show these relations is the following:

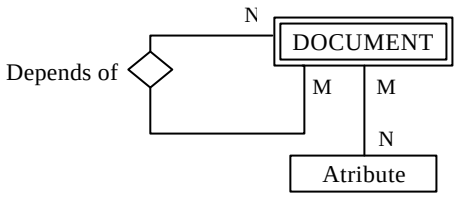


Figure 30. E/R for the attribute of documents and relations between documents

If we join all this entities and relations together we obtain the E/R's diagram of the whole database.

The database consists of 9 tables. Each one stores information about a particular type of entity (we have split the information in tables for a better functioning of the database, for to improve the actualization of data and the celerity of the consultations).

All the information relating to a generic project is stored in a database. There are several tables in the database design, as shown in Figure 32. One of these tables is called 'Relation' and enables the different information from each table to be matched up.

The fact that the same actor can assume different roles in different projects compels us to define all the potential actors/roles related to a document.

In general terms, the information is divided into the following tables:

- PHASE and STAGE, where the previously defined different phases and stages along the project life cycle are stored.

- ACTIVITY and SUBACTIVITY, where the various project areas are stored.

- AGENT, that stores the three main agents of a project: Client, Designer and Contractor.

- ROLE, that stores information about the role each project agent can perform. It's important to notice that one agent can carry out many different roles, and one role can be played by different agents.

- DOCUMENT, that contains all the metadata explained above, such as document name, description, late submission date, revision number, and type of stored information.

- ATRIBUTE is an auxiliary table that gives extra information about the document, such as the format of the document.

- RELATION, that contains the organization of documents in to Phases / Stages /Activities / Subactivities / Documents, and tells who creates and receives these documents and which documents are concerned.

- CONTRACT assigns each actor a task related to a document. It consists of a type of contract and their characteristics. It is used to select the responsibilities of an agent/role for a document.

- CREATE-RECEIVE contains the responsibilities of each agent depending on the contract, the type of document and the place of this document in the life cycle of the project.

Following the E/R diagram called Chen's diagram is exposed.

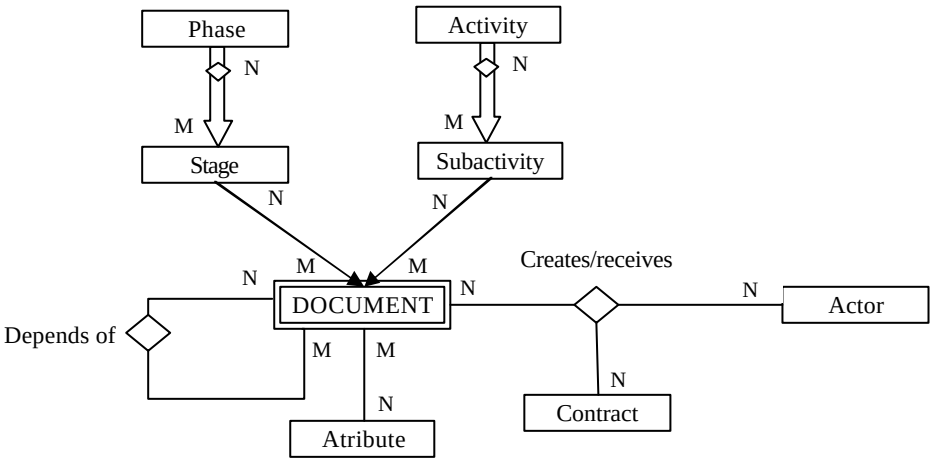


Figure 31. E/R diagram of the proposed database

Phase	Stage	Activity	Subactivity	Document
name phase	name stage	name activity	name subactivity	doc name project revision nº description

Type	Contract	Actor	Role
type name extension	type of contract characteristics	name actor	type of role company address contact person

Figure 32. Contents of different tables

Figure 33 shows the shows the proposed conceptual schema of the system, broken down into three different dimensions: Presentation, Product and Life cycle dimension.

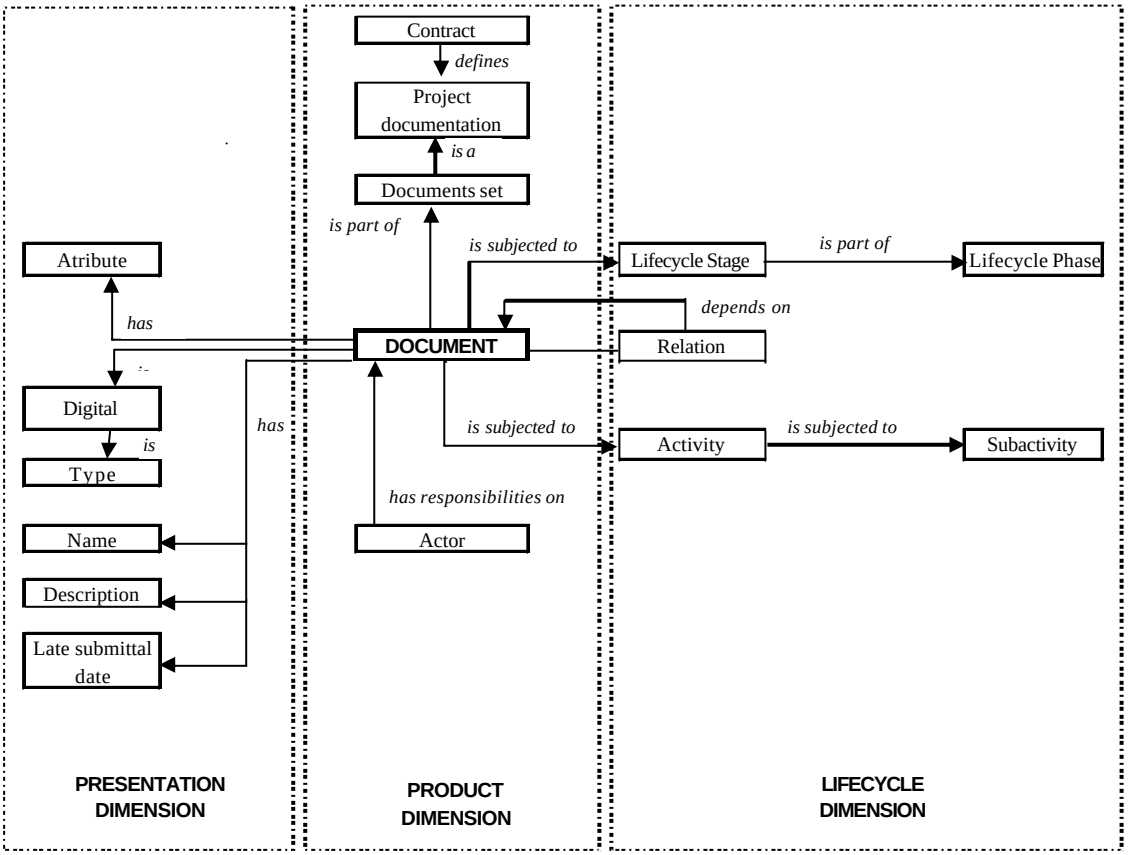


Figure 33. Conceptual schema of the System

9.4.1. Contents of each table

phase		
Id phase	**phase**	**fase**
can	Conception	Concepción
dn	Desactivation	Desactivación
ec	Execution	Ejecución
cc	Purchase and contractings	Compras y contrataciones
dt	Technical Design	Diseño Técnico

stage		
Id stage	**stage**	**etapa**
apr	Conceptual Definition	Definición conceptual
con	Construction	Construcción
cde	Construction Delivery	Entrega
cnt	Contractings	Compras y Contrataciones
dpr	Detailed Proposal	Propuesta detallada
sbr	Feasibility Study	Estudio de viabilidad
fpr	Final Proposal	Propuesta final
mai	Maintenance	Mantenimiento
opr	Outline Proposal	Propuesta inicial
pcc	Production of information for contractings	Producción de información para contratación
pot	Production of information for tender	Producción de información para licitación
tac	Tender action	Concurso o licitación
tdo	Tender documentation	Documentación para concurso o licitación

activity		
Id activity	**activity**	**actividad**
adv	Advance	Avance
chn	Changes	Cambios
cnt	Contractings	Contrataciones
cst	Costs	Costes
env	Environment	Medio Ambiente
prg	Programming	Programación
prj	Project	Proyecto
qlt	Quality	Calidad
rsg	Risks	Riesgos
sah	Safety and Health	Seguridad y Salud

subactivity		
Id subactivity	**subactivity**	**subactividad**
com	Comunication	Comunicación
doc	Documentation	Documentación
log	Logistics	Logística
mac	Monitoring and Control	Seguimiento y Control
cdn	Organization	Organización
pln	Plan	Plan
pur	Purchasing	Compras, Suministros
std	Studies	Estudios

document		
Id document	**document**	**documento**
tad	Administrative procedure	Trámite administrativo
ann	Annexe	Anejo
rps	Answer	Respuesta
afd	Approval	Aprobación
dac	Awarding	Adjudicación
pre	Budget	Presupuesto
oes	Bylaws	Ordenanzas, Normativas y Reglamentos
cit	Catalogue	Catálogo
crt	Certification	Certificación
cmb	Change order	Orden de cambio
idc	Communication report	Informe de Comunicación
cnt	Contract	Contrato
pcc	Control Plan	Plan de Control
ldd	Defects List	Lista de defectos
pls	Drawing	Plano
fmt	Form	Formato
dgc	Generic document	Documento genérico
inc	Incidence	Incidencia
sif	Information request	Solicitudes de información
pdi	Inspection Points	Puntos de inspección
int	Instruction	Instrucción
seg	Insurance	Seguro
alb	Delivery note	Albarán
inv	Invoice	Factura
cts	Letter	Carta
lcp	Licence	Licencia-permiso
man	Manual	Manual
med	Measurements	Mediciones
reu	Meeting	Reunión
mem	Memory	Memoria
act	Minute	Acta
ncf	Non-conformities	Inconformidades
org	Organization chart	Organigrama
pgs	Payments	Pagos
plf	Planning	Planificación
prc	Procedure	Procedimiento
prg	Programming	Programación
rec	Receipt	Recibo
ins	Report	Informe
pdo	Request of tender	Petición de ofertas
ldo	Site Book	Libro de obra
snt	Site Note	Nota de obra
spc	Specifications	Especificaciones
orb	Tenders	Ofertas
fct	Turnover	Facturación
tcl	Visa procedure	Trámite colegial

agent		
Id agent	**agent**	**agente**
i	Client	Cliente
o	Contractor	Contratista
d	Designer	Diseñador

contract		
Id contract	**contract**	**contrato**
pc	professional construction management arrangement	contrato dirección integrada
tr	traditional procurement arrangement	contrato tradicional
tk	turnkey project arrangement	contrato llaves en mano

9.5. Implementation of the database to a web based system

9.5.1. Possible solutions

Microsoft Office is the most popular and the easiest software for any particular or SME that cannot invest on tools with major possibilities neither has enough knowledge about more complicated tools.

On the other hand, Internet evolution has lead to new technologies. The contents first were totally static but in a short period since the arrival of scripting languages, the World Wide Web together with HTML language has reached to dynamic pages. In these dynamic pages the user cannot only read, search and browse information, but also interact with this information and send it via forums, forms, quests, comments, etc.; the page administrator can give permissions to users, so avoiding constant modifications of the page base code.

According to Trigos (2000), 90% of Internet websites implement a database system to actualize, modify, eliminate and incorporate information. DMS or WPMS are based on these dynamic pages.

PHP is one of the languages broadly used by programmers due to its vast advantages, like being Open Source, and to its broad functioning platforms: Windows, Unix, Solaris, etc. Some databases that can work with PHP are SQL Server, MySQL, etc. MySQL (My Structured Query Language) is a Relational Database Management. As aforementioned, a database is a structured collection of data, and a relational database is characterized for having all these information in tables so the relations among data are made explicit in these same

data. This increases speed and flexibility. MySQL is also an Open Source software, fast, robust, and easily usable for large and small projects.

It's obvious that if the model we are proposing is to be used to improve the document organization of those companies working together in a WPMS, it's necessary to have the database available on Internet. In this case, users don't need to download any type of software, but just access Internet and choose the specific information such as the type of contract and the type of actor. The results will be automatically viewed locally (in the user's PC) or in the WPMS.

We have developed the database in MySQL using PHP to upload the database via Internet, but also in Access, in case Internet access is not possible and users only want to use this database locally.

9.5.1.1. My SQL

MySQL database server is the world's most popular open source database. MySQL has quickly become the core of many high-volume, business-critical applications.

Customers such as Yahoo, Google, Cisco, Sabre Holdings HP and NASA, are realizing significant cost savings by using MySQL's high performance, reliable database management software to power large Web sites, business-critical enterprise applications and packaged software applications.

MySQL offers several key advantages:

- **Ease of Use and Deployment.** MySQL's architecture makes it extremely fast and easy to customize. Its multi-storage engine architecture gives customers the flexibility they need with a DBS such as speed, compactness, stability and ease of deployment.

- **Open source.** This refers to a Free Software-related kind of software licensing and distribution. The consumer of an open source program is entitled to: read, use, modify and distribute the source; charge money for services, copying or support, so long as they do not hinder others' freedom. Open source isn't PublicDomain. It means that there is a license involved, and the license has restrictions which can include: distribution must be free; modifications must be distributed; original authors must be acknowledged; derivatives must be similarly licensed.

- **Cross-Platform Support.** MySQL is available on more than twenty different platforms including major Linux distributions, Mac OS X, UNIX and Microsoft Windows.

- **Millions of Trained and Certified Developers.** MySQL is the world's most popular open source database, so it's easy to find high-quality, skilled staff.

MySQL has the capabilities to handle most corporate database application requirements with an architecture that is extremely fast and easy to use.

9.5.1.2. PHP

PHP, which stands for 'PHP: Hypertext Preprocessor', is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly.

9.5.2. Application of the database to a web product

The contribution of this thesis is the automatic creation of the organization of folders, subfolders and archives before starting using a WPMS or before starting using an EDMS.

This system can be used alone, before starting a project, with the aim to create the same folder structure for all the actors taking part in a project. All the actors should access the system and download the folder structure for the specific project created automatically. It can also be implemented as a function of a WPMS. Once the different actors access the WPMS they can have the functionality of downloading the same folder structure previously created by the web system. This model is designed to be used by any SME or any kind of WPMS.

Before starting using a WPMS, the organization of folders, subfolders and archives should be designed. For this reason, this model is independent of the WPMS in use: Foreign companies with ICT services; ICT companies with or without national support that operate in the same country; Construction Companies that have created their own EDMS via extranet.

Current WPMS provide the same services such as chat, webcam, document version management, agenda, news, archive management, etc.

When a company considers using a WPMS for a specific project or for all its projects, an actual DMS can be used or the company can create its own. However, none of the existing software in Spain maps all the possible documentation generated throughout the life cycle of the project. So, the system is a web service where to define the characteristics of the project and to obtain the organization structure of documents to be used in each specific project.

Then, the application will allow to create a local folder organization of documents or to create it in whatever WPMS.

A web based 'Life cycle Document Management System' was created to test the database model. It can be accessed online at http://www.constructiondms.upc.es.

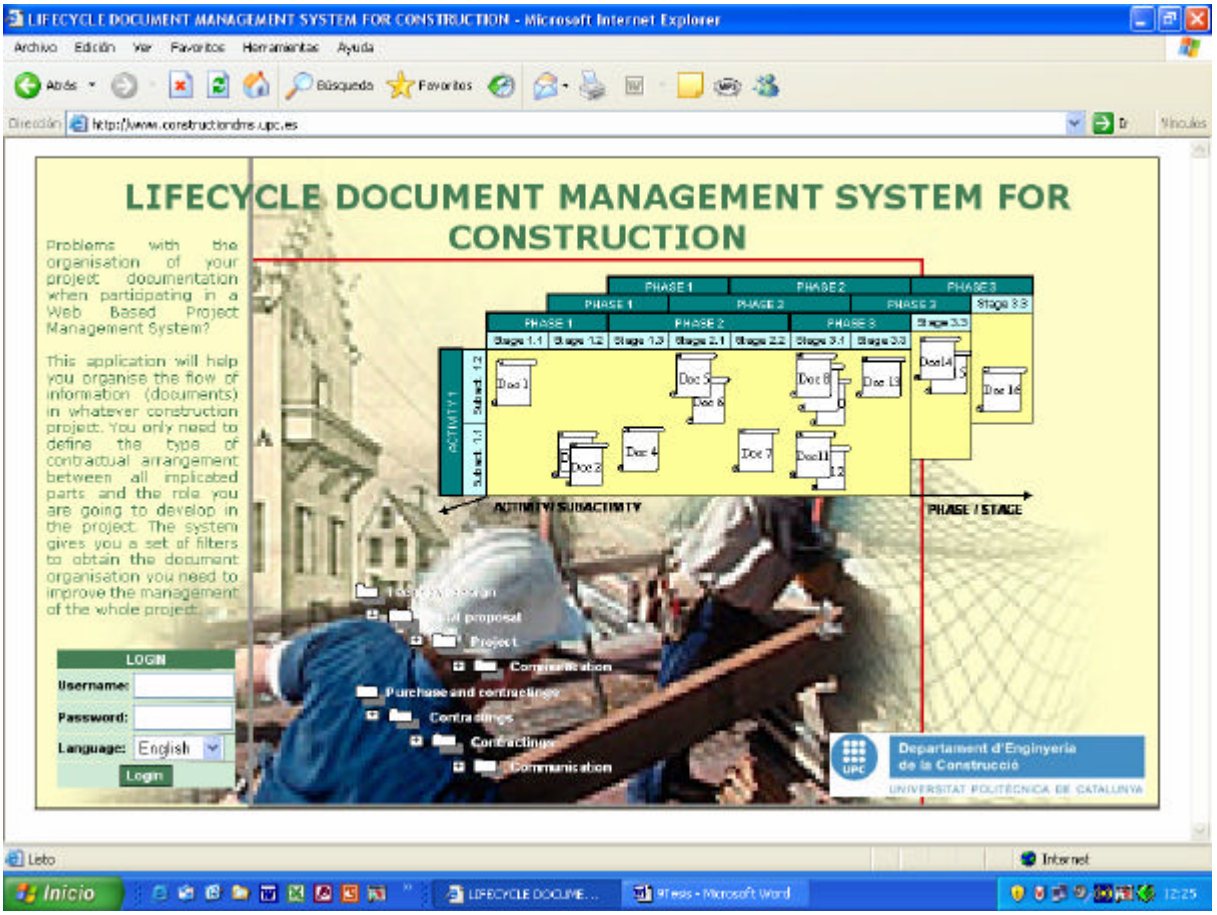


Figure 34. Access to the Construction Document Management System

The previous figure shows the access web page to the System. For the moment, the system is available in English and Spanish but it's a language independent tool, so it can be easily translated into whatever language.

The *User's guide* is available in *Appendix II.* Notwithstanding, the system is self understandable, in each screen the user will find the purpose of the system and an explanation of what to do in each step.

The Administrator should be aware of all the participants and their roles throughout the entire project, and will be in charge of ensuring that the web page functions correctly, and of managing the project.

Once the Administrator has introduced his/her username and password, he/she should create a project, define the specific contractual arrangement for this project and assign passwords to the users (actors) of the specific project. As aforementioned, one user can have different roles in different projects, so each role will be chosen and assigned to the user.

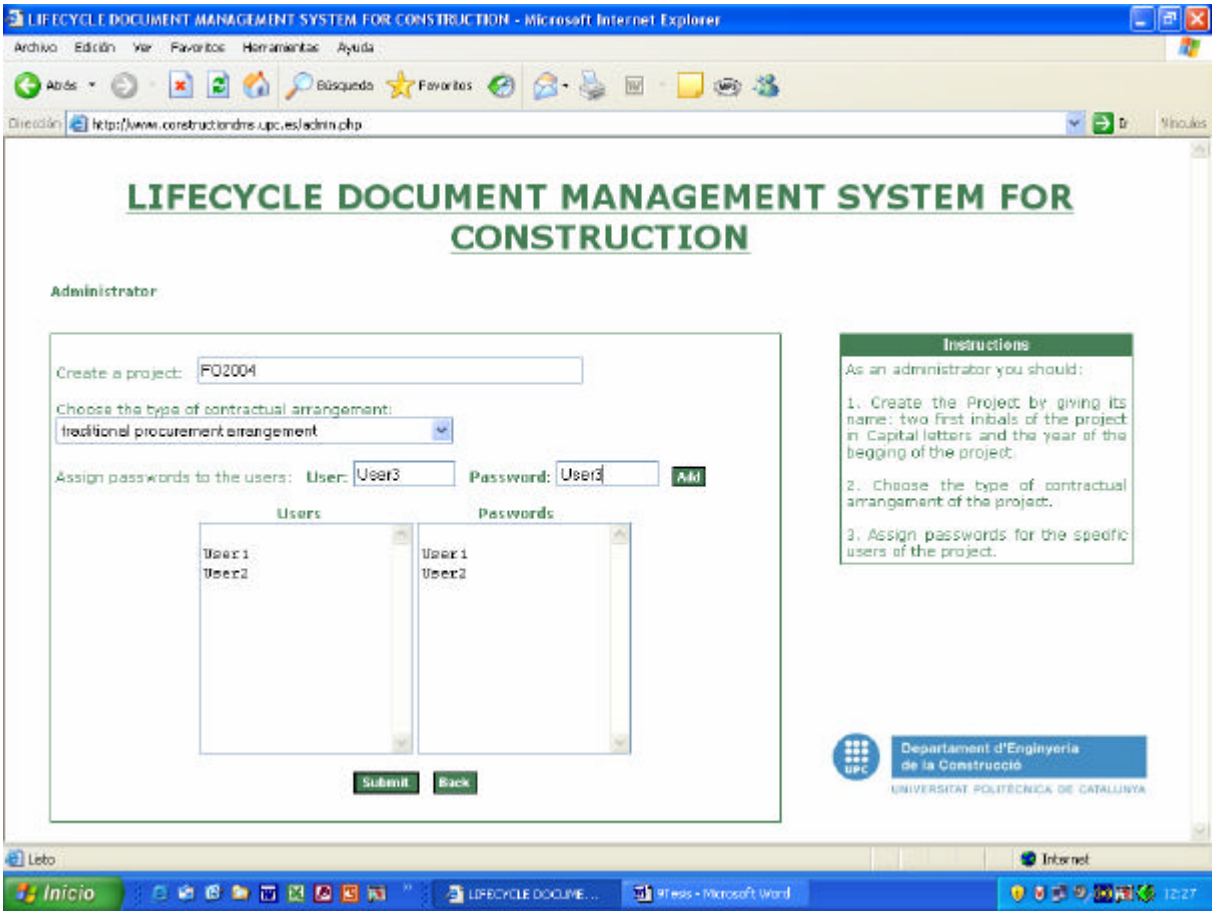
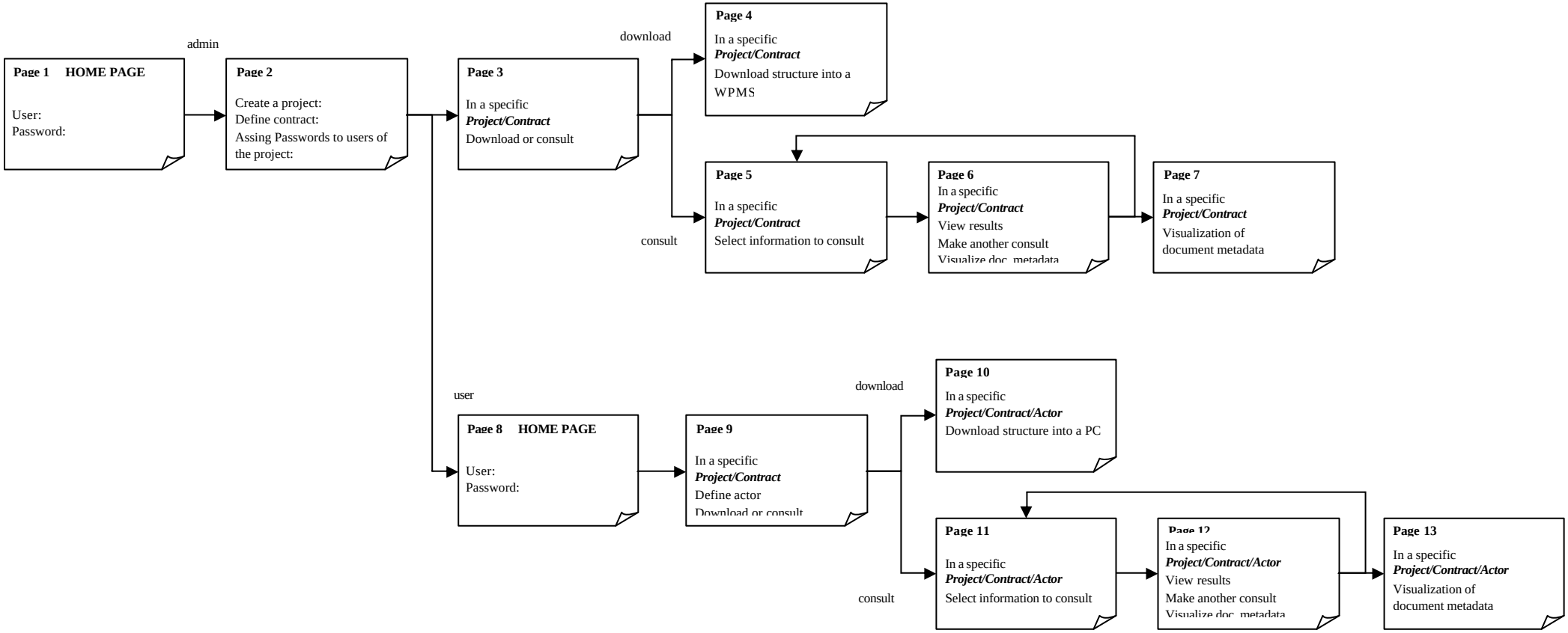


Figure 35. Assignment of Users and Passwords

Figure 35 shows the first screen, where the Administrator creates a project, chooses the type of contractual arrangement and assigns passwords to users.

Once the users have their password they can access the system. The next figure shows the general structure of the system, organized by screens and functionalities.


admin
download
Page 1 HOME PAGE
User:
Password:
Page 2
Create a project:
Define contract:
Assing Passwords to users of the project:
Page 3
In a specific
Project/Contract
Download or consult
Page 4
In a specific
Project/Contract
Download structure into a WPMS
consult
Page 5
In a specific
Project/Contract
Select information to consult
Page 6
In a specific
Project/Contract
View results
Make another consult
Visualize doc. metadata
Page 7
In a specific
Project/Contract
Visualization of document metadata
user
download
Page 8 HOME PAGE
User:
Password:
Page 9
In a specific
Project/Contract
Define actor
Download or consult
Page 10
In a specific
Project/Contract/Actor
Download structure into a PC
consult
Page 11
In a specific
Project/Contract/Actor
Select information to consult
Page 12
In a specific
Project/Contract/Actor
View results
Make another consult
Visualize doc. metadata
Page 13
In a specific
Project/Contract/Actor
Visualization of document metadata

After accessing the system, the administrator and the actors will have the same possibilities, except for downloading the folder structure in a WPMS, which will only be accessible for the administrator.

Once the user has the password for a specific project, he/she should define the type of actor he/she is, i.e. client, contractor and/or designer. For each specific project there is a specific organization of documents related to the contractual arrangement and to the role each user is playing.

The user, then, has two options: Download or Consult.

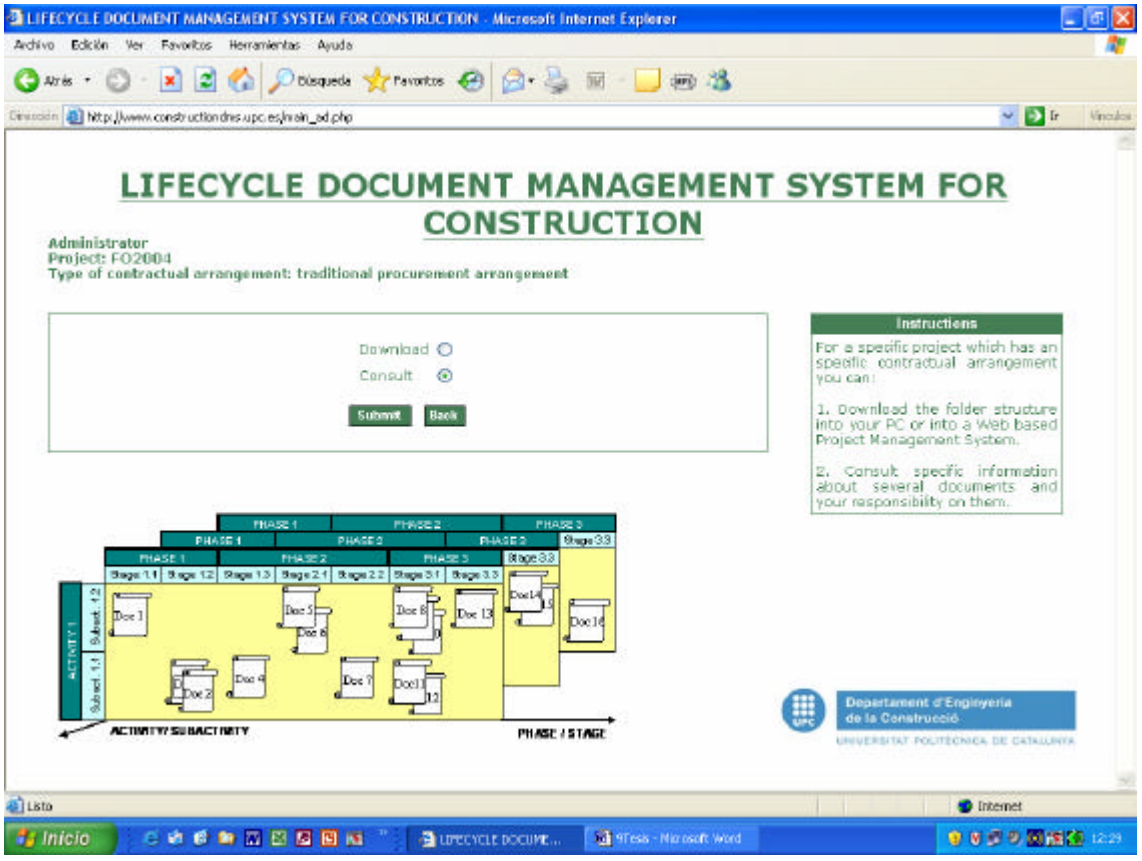


Figure 36. Options of the System

With the Download option, the user can download into his/her PC the same folder structure, which the administrator has already uploaded into the WPMS. In this case, the system creates a life cycle folder structure with all the documents to be used in the project.

The following figure shows the ‘Download option’.

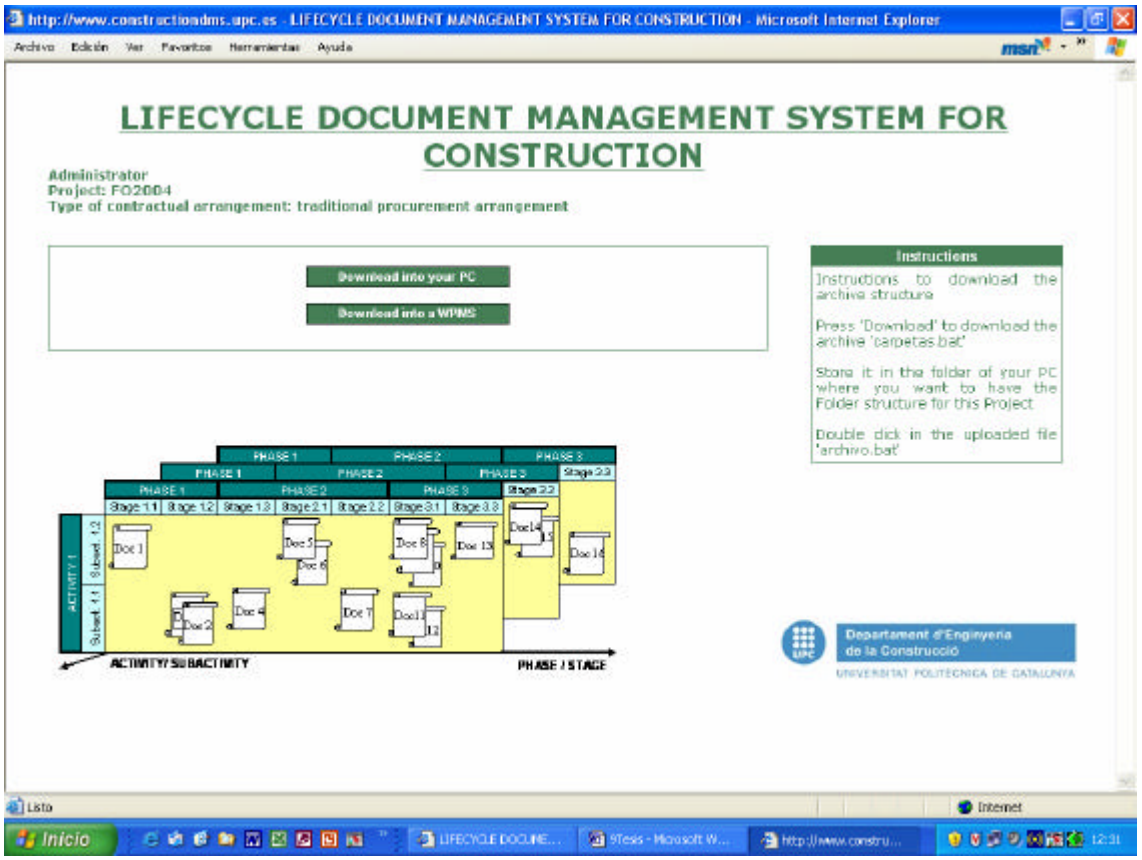


Figure 37. Options to download the folder structure

After choosing the option ‘Download in your PC’, if you double click on the archive ‘Carpetas.bak’ the system creates the specific folder and documentation structure for the project. The following screen shows an example of a project FO2004 with its folder structure and the documents of a subfolder.

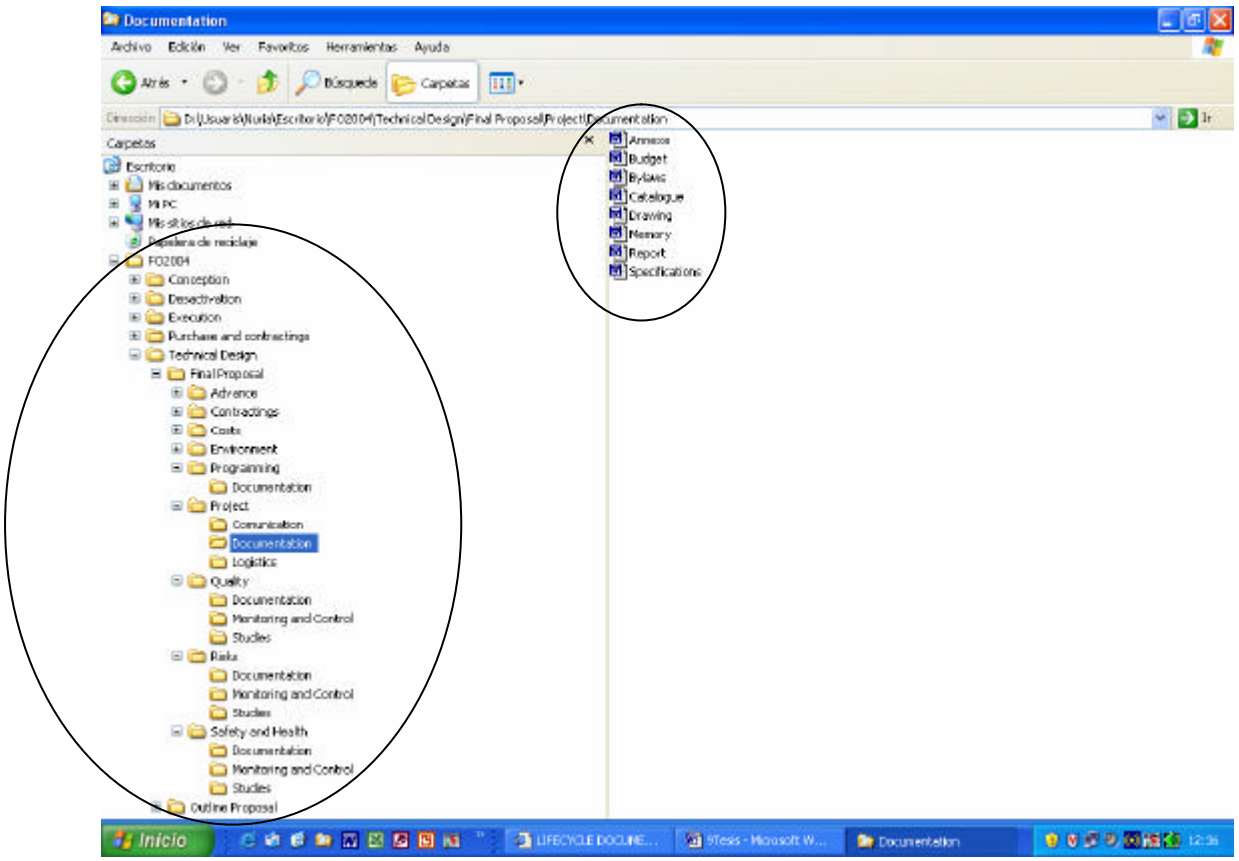


Figure 38. Creation of the folder structure

Another option is to make consults about specific documents to be uploaded or downloaded in a specific project. The following figure shows the screen used to select information such as: type of document, responsibilities -which means if the actor should create that document and upload it into the WPMS, or if he/she will receive it and use it. The other information is organized in a matrix where the user can choose the phases and stages along the life cycle, the activities and subactivities.

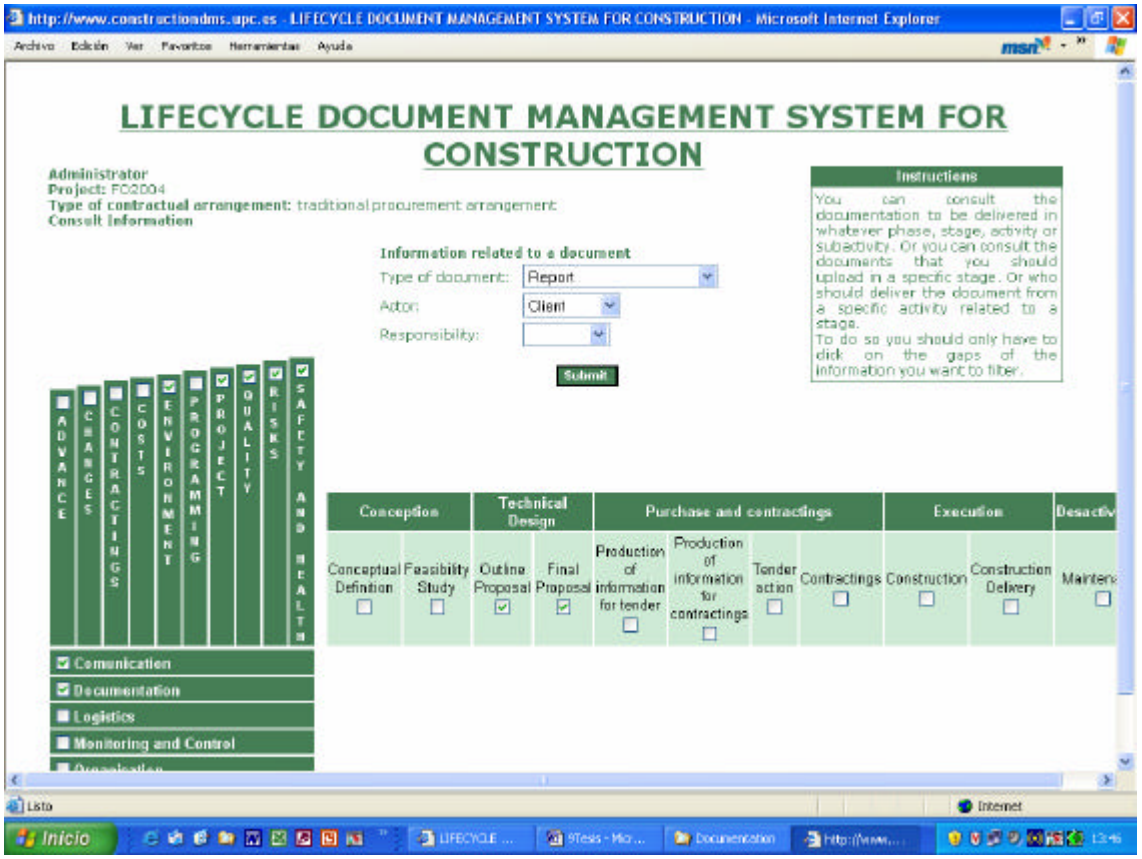


Figure 39. Screen for making consults

The actor can consult the documentation to be delivered in whatever phase, stage, activity or subactivity. Or he/she can consult the documents that he/she should upload in a specific stage. Or who should deliver the document from a specific activity related to a stage.

To do so, actors should only have to click on the gaps of the information they want to filter.

After choosing the information to be filtered, the system returns a table like in the following Figure, with the results.

Figure 40. Results of the consult

If the user wants more information about a document, he/she should click on it, and the system returns all the information related to that document. An example of that screen is shown in the next figure.

Figure 41. Metadata of a specific document

9.6. Database functioning

Although database can be used for many different purposes, the most important function consultation.

Consults are basically done through the 'Relation' table, but they can also be done through any other table.

We can limit the fields to be shown. For example, if we choose the 'turnkey contract' and the actor 'constructor', we can limit the consult and ask only to give us the 'Phases', 'Stages' and 'Activities' where this actor should create a 'Minute'. Then we would obtain the following results:

Table 19. Results obtained from the proposed database when we select: Turnkey contractual arrangement where the contractor will have to create a Minute.

Phases	Stage	Activity	Subactivity
Technical Design	Final Proposal	Project	Communication
Purchase and contractings	Contractings	Contractings	Communication
Execution	Construction	Advance	Organization
Execution	Construction	Changes	Organization
Execution	Construction	Changes	Communication
Execution	Construction	Contractings	Communication
Execution	Construction	Costs	Communication
Execution	Construction	Environment	Communication
Execution	Construction	Programming	Organization
Execution	Construction	Project	Communication
Execution	Construction	Quality	Organization
Execution	Construction	Risks	Communication
Execution	Construction	Safety and Health	Communication
Execution	Construction Delivery	Programming	Communication
Desactivation	Maintenance	Advance	Communication
Desactivation	Maintenance	Costs	Communication
Desactivation	Maintenance	Environment	Communication
Desactivation	Maintenance	Programming	Communication
Desactivation	Maintenance	Project	Communication
Desactivation	Maintenance	Quality	Communication
Desactivation	Maintenance	Safety and Health	Communication

Then this table will be translated into:

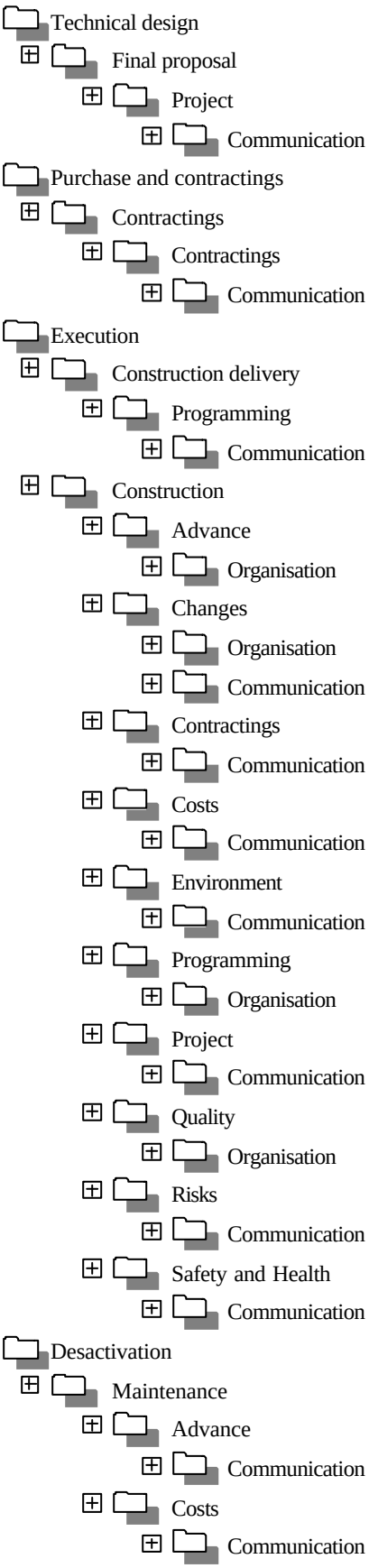


Figure 42. Translation of the results into an organization document file

We can also obtain the results of all the documentation to be created or received by any actor in a 'typical contractual arrangement', from defining a specific Stage.

Table 20. Results obtained from the proposed database when we ask to obtain all the documentation to be created or received by any actor in a typical contractual arrangement, from defining the Conceptual definition stage and the Advance and Project Activities.

Activity	Subactivity	Document	Create	Receive
Advance	Monitoring and Control	Generic documents	Designer	Client
Advance	Monitoring and Control	Reports	Designer	Client
Advance	Communication	Minutes	Designer	Client
Advance	Communication	Letters	Designer	Client
Advance	Communication	Communication reports	Designer	Client
Advance	Communication	Meetings	Designer	Client
Advance	Communication	Answers	Client	Designer
Advance	Communication	Information requests	Client	Designer
Project	Documentation	annexes	Designer	Client
Project	Documentation	Memory	Designer	Client
Project	Documentation	Reports	Designer	Client
Project	Communication	Information requests	Client	Designer
Project	Communication	Minutes	Designer	Client
Project	Communication	Letters	Designer	Client
Project	Communication	Communication reports	Designer	Client
Project	Communication	Meetings	Designer	Client
Project	Communication	Answers	Client	Designer

9.7. Example

A Sports company (CLIENT) decides to build a new Sports Centre in Barcelona. To do so, it contracts 'an Architect (DESIGNER) to design the building. This architect subcontracts an engineer (DESIGNER) for the specific HVAC and electrical facilities.

The Client has a direct relationship with some CONTRACTORS because it's not the first Sports Centre it opens, so it decides to contract a main contractor who will be responsible for the construction itself, even though the work will be actually undertaken by different specialties subcontractors.

When the project starts, the Client firm wants to hold direct relationship with most participants and to control their work. Because of this, it decides to contract a WPMS-ASP where all the stakeholders of the project should store their information. The client decides to outsource this service because he is not specialist on IT. The main functionalities of these WPMS are Team Communication and Document Management. The company decides to use a WPMS because it is convinced that the communication

between all the stakeholders will improve, reducing costs, but the most important point is the improvement of project control.
Before starting using this WPMS, as it doesn't provide a folder structure where to store all the information, the client decides to use the 'Life cycle Document Management System for Construction'. With this tool, the administrator, in this case the Client defines the type of contractual arrangement and the users.

Type of contractual arrangement: Traditional procurement arrangement
User 1: Client
User 2: Architect (Designer)
User 3: Engineer (Designer)
User 4: Main contractor (Contractor)
User 5: Electrical Supplier (Contractor)
User 6: HVAC installer (Contractor)

This System generates the folder, subfolder, documentation, etc., for the Sports Centre project. It defines 3 actors: The Client, the Designer which involves the architect and the engineer, and the Contractor involving the main contractor, electrical supplier and HVAC installer.

The administrator downloads into the WPMS the folder structure provided by the 'Life cycle Document Management System'.

Now, when accessing the WPMS, all the partners will view the same folder structure and they will probably know where to store all the information.

The administrator also provides passwords to the 'Life cycle Document Management System' to all the users. These users will only have access to the folder structure that the administrator has created for the specific project. Then each actor will be able to download this folder structure to their PC, server or Intranet.

Now, all the actors of the project will have internally the same folder structure as the WPMS-extranet, and it will be easy to know where to store each piece of information and where to find it if needed.

Different parts of the project life cycle will be shown as an example of functioning.

1. The Client provides the specifications. Instead of printing and giving them to the designer, or sending them via email, the client uploads the specifications into the correct folder and subfolder of the WPMS.

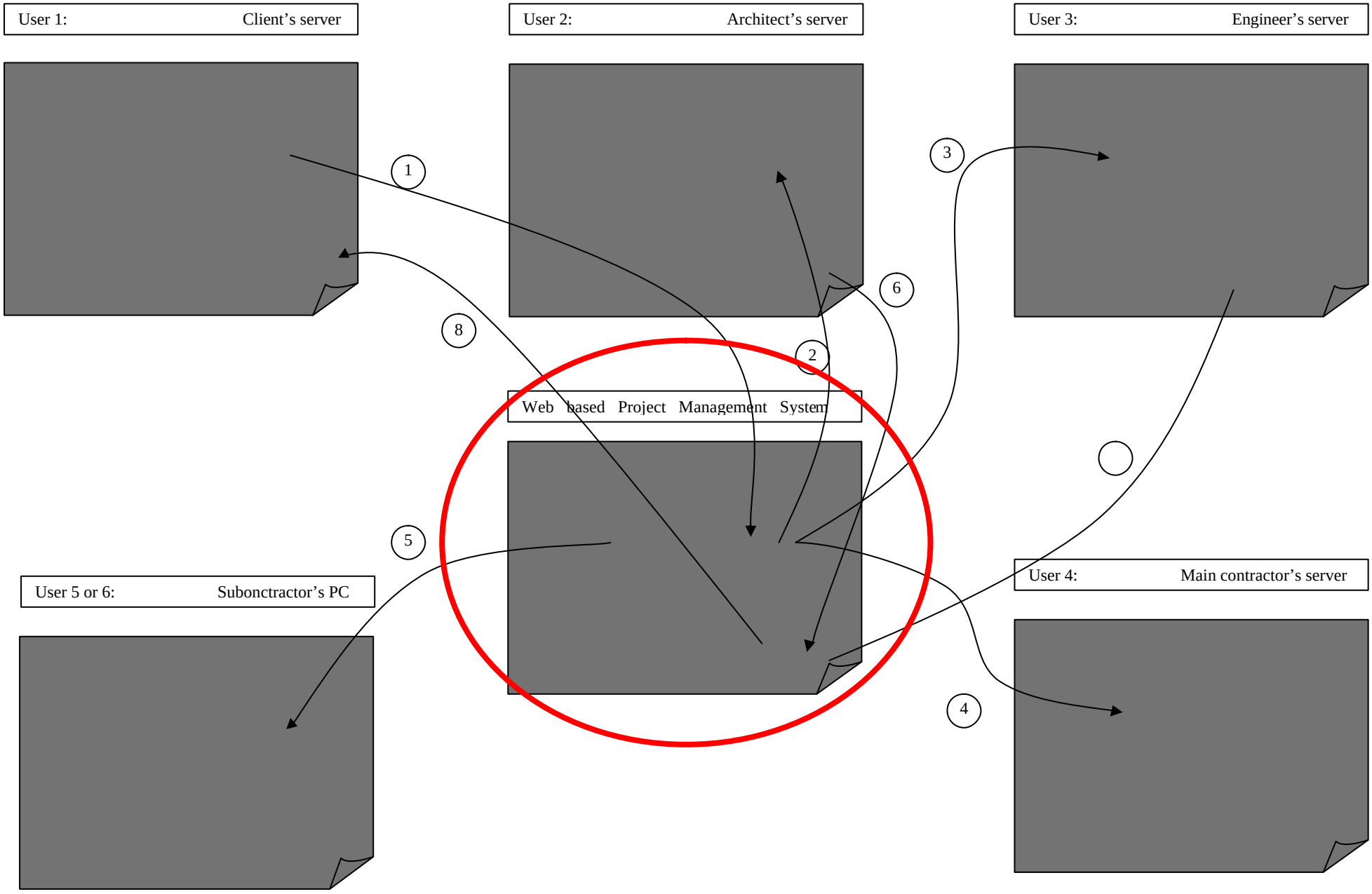
2. The architect accesses the WPMS and downloads the specifications so as to work with the proposal.

3. The architect needs extra information from the engineer, so he asks the engineer to access the WPMS and work with the proposal uploaded by the client.

4/5. The architect must deliver a programming of works, so he ask the Main Contractor, who at the same time asks all the implicated subcontractors, to access the specifications and other relevant information, and to provide a possible timing.

6. The client asks the architect to deliver a programming of works. He works with it and stores his documentation in his server but in the correct folder and subfolder. Then, when delivering the documentation, he only has to upload the document into the WPMS and into same folder to his own server.

7/8. Then, both the client (to control the project) and the engineer (to be aware of the programming of works) have access to this information.



LIFE CYCLE DOCUMENT MANAGEMENT SYSTEM FOR CONSTRUCTION

9.8. Summary

This web based Life Cycle Document Management System for Construction is based on the real necessities of some big companies working with WPMS and of those SMEs that in the near future will be obliged to use these tools.

With this system, an easiest access to information and exchange of documentation will be provided to all the actors taking part in a project.

All relevant project information such as Phase, Stage, Activity, Subactivity, Actor, Role, type of document, attribute, contractual arrangement, and other document metadata previously defined and linked in Chapter 7, is also taken into consideration.

Currently, Spanish and English versions are available in the web but, being a language independent tool, it can be easily translated into whatever other language.

The *User's guide* has been also created to solve problems of misunderstandings or lack of information.

This model is in continual development, based on the underlying premise that new project organization and management frameworks may help work practices better fit the emerging technological tools.