

POLITECNICO DI TORINO

III Facoltà di Ingegneria
Corso di Laurea in Ingegneria informatica

Tesi di Laurea Magistrale

Analysis, design and development of an Informative System deploying RFID technology



Relatore:

prof. Maurizio Rebaudengo

Nicola ADAMO
Fabio AULICO
Giuseppe CASTELLUCCI

ANNO ACCADEMICO 2008-2009

Contents

I State of the Art of RFID technologies and early phases of project development	1
1 Introduction	2
1.1 Purpose of the thesis	2
1.2 IT and research field interested by our project	2
1.3 The idea and case study	4
1.4 Approach to the project	4
1.5 Document structure	5
2 State of the art of RFID Systems	6
2.1 RFID: what, when, why	6
2.2 The technology	8
2.2.1 The RFID system	9
2.2.2 Frequencies and standards	13
2.3 Typical applications	17
2.3.1 Animal identification	18
2.3.2 Logistics	18
2.3.3 Access control	20
2.4 Security and privacy issues	21
2.4.1 Privacy issues	21
2.4.2 Security issues	22
II From the idea to the development	25
3 The idea	26
3.1 Introduction	26
3.2 Features of leisure time and touristic structures	26
3.2.1 Role of the ICT and BPR in Corporates	27
3.2.2 Introduction to a specific application context: touristic resorts, restaurants, parks, sports centers	28
3.3 OUR idea	30
3.3.1 Features	30

3.3.2	The RFID tag equipped accessory	30
3.4	Discovering and analysis of application context requirements	31
3.4.1	Check-in desk procedure	31
3.4.2	Information/help desk points	33
3.4.3	Goods and services selling points	34
3.4.4	Monitoring structure activities	34
3.5	Market competitors	35
3.5.1	The research	36
3.5.2	F.C. Consulting Group: SummerMagic	36
3.5.3	WSC - World System Consulting: Quick Suite	38
4	Application case study	40
4.1	Introduction	40
4.2	Focusing on the beach resort	40
4.3	Our approach to process reengineering	41
4.3.1	Project hypothesis and Feasibility Study	41
5	System analysis and design	58
5.1	Overview	58
5.2	Approach to Requirements analysis	58
5.2.1	Emerged functionalities	60
5.2.2	Bar module	68
5.2.3	Turnstiles	68
5.2.4	PDA services module	68
5.2.5	Non functional requirements	71
5.3	System design	75
5.3.1	Box office	75
5.3.2	Remote interface	82
5.3.3	Bar payment module	85
5.3.4	Turnstiles management module	92
5.3.5	IT Kiosk	96
5.3.6	PDA services module	102
5.4	Hardware	106
6	System development	108
6.1	Overview	108
6.1.1	Presentation, business and data services	108
6.2	Data layer	110
6.2.1	Overview	110
6.2.2	Mapping relational data to business entities	112
6.2.3	Object-relational mapping	115
6.2.4	.netTiers	118
6.2.5	Step-to-step data layer creation	121
6.3	Business layer	124

6.3.1	Overview	124
6.3.2	Business Layer Implementation Approach	124
6.3.3	Overall Business layer structure	125
6.4	Presentation layer	127
6.4.1	Overview	127
6.4.2	User end issues	129
6.4.3	Architecture issues	130
6.5	System decomposition	132
6.5.1	Box office	132
6.5.2	Remote interface	149
6.5.3	Bar payment module	156
6.5.4	Turnstile management Module	161
6.5.5	IT Kiosk	163
6.6	Networking	168
III	Analysis of results and project closure	171
7	Analysis of results	172
7.1	Overview	172
7.2	Bar payment analysis	172
7.3	Virtual account recharge analysis	173
7.4	Check-in execution mean time	177
7.5	Tag losses trend	178
8	Project closure and future development	182
8.1	Post-completion project overview	182
8.1.1	Achievement of project expected results	182
8.1.2	Errors and successes on fundamental project activities	185
8.1.3	What were the lesson learnt	187
8.2	Future developments	188
8.2.1	Missing modules and functionalities	188
8.2.2	Data mining	188
8.2.3	Customer profiling	188
8.3	Conclusions	188
A	Riassunto in lingua italiana	190
A.1	Introduzione	190
A.1.1	Scopo della tesi	190
A.1.2	Contesto della tesi	190
A.1.3	L'idea ed il case study	191
A.1.4	Approccio al progetto	192
A.2	Stato dell'arte	192
A.2.1	Perchè RFID?	192

A.2.2	Il sistema RFID	192
A.2.3	Frequenze	194
A.2.4	Standard	194
A.2.5	Applicazioni tipiche	194
A.2.6	Problematiche di sicurezza e privacy	195
A.3	L'idea	195
A.3.1	Analisi del contesto applicativo: resort, villaggi, poli fieristici, parchi	196
A.3.2	La nostra Idea	196
A.3.3	Analisi del contesto applicativo	197
A.3.4	Procedura di check-in e controllo accesso	197
A.3.5	Desk informativi	198
A.3.6	Punti vendita di beni e servizi	198
A.3.7	Monitoraggio della struttura	199
A.3.8	Il mercato: concorrenti	200
A.4	Il case study	200
A.4.1	Introduzione	200
A.4.2	Applicabilità ad una spiaggia attrezzata	200
A.4.3	La nostra reingegnerizzazione dei processi	201
A.4.4	Identificazione dei processi e raccolta dei requisiti	204
A.5	Analisi e design del sistema	204
A.5.1	L'analisi dei requisiti	205
A.5.2	Design del sistema	207
A.6	Sviluppo del sistema	208
A.6.1	Layering	208
A.6.2	Considerazioni sul Data Layer dell'intero sistema	209
A.6.3	Decomposizione del sistema	209
A.6.4	Modulo di pagamento al Bar	211
A.6.5	Chiosco informativo	213
A.6.6	Modulo di gestione tornelli	214
A.7	Analisi dei risultati	215
A.8	Chiusura del progetto e sviluppi futuri	216
A.8.1	Raggiungimento dei risultati attesi di progetto	216
A.8.2	Lezioni imparate	217

Part I

State of the Art of RFID technologies and early phases of project development

Chapter 1

Introduction

The hereby presented project has been conducted in order to accomplish the final step of the Master of Science in computer engineering at the Technical University of Turin, as it represent the master thesis of the three candidates stated in the title page of this document.

1.1 Purpose of the thesis

The purpose of our work is to realize a project, from the idea to the real installation, of an Informative System (IS) deploying RFID technology able to support the management of recreative/commercial structures (amusement parks, exhibition fairs, sky resorts and so on). The system aims to automate the processes typical of this kind of structures and to arrange an information framework on which the management can base short, medium and long term strategic decisions. A wearable device (tag) based on RFID technology, constitutes the mean for visitors/clients identification: the punctual identification of the single person becomes an enabling factor for all the functionalities intended for the realized system. Furthermore, we want to evidentiate the structured approach to project organization and decomposition and a development methodology that follows the standards and best practices for IS implementation receiving the larger consensus by the international scientific community.

1.2 IT and research field interested by our project

The chance for an RFID based informative system arise from different factors:

- Substantial decrease in RFID technologies costs;
- Wide convergence on RFID standards;
- Business empowerment possibilities related to the adoption of RFID technologies;
- High level of reliability, robustness and durability of RFID components;



Figure 1.1. The wearable RFID tags involved in our project

and much more... The knowledge developed in this relatively new field of technology is posing a base layer for application developers who want to create software solutions capable to catch all its potentials. On the other side, these potentials, together with a sensible reduction of the necessary investment, push an increasing number of companies towards the introduction of IT solutions able to take advantage of them. With the growth of RFID application as a support to supply chains, warehousing and goods' tracking, a new "Internet of things" is about to become a reality. While this tendency is bringing the adoption of solutions targeted on logistics, transportations and animals identification to a level of relative maturity, the same cannot be said for those solutions based on people identification. Furthermore, the extent of usage in people's everyday life has encountered several criticism based on security and health preservation issues. Lately, in most of the developed countries, a government driven regulation process is providing the means for a "careful" implementation of RFID based systems. The extent to which these are spread, especially in Italy, is quite low, and this is bringing companies and researchers interested in RFID technology to focus their attention on those countries that have already taken advantage of RFID opportunities. Estonia can be included among these countries. The analysis, design and implementation of the system took place in Tallinn, where we got the support of Eliko Competence Center, a company founded by Tallinn Technical University (along with some commercial partners) that aims to "improve competitiveness of Estonian and European ICT industry through the deep integration of research and production institutions. ELIKO will enable reducing risks of development and validation of emerging ICT solutions for tomorrow". Among these emerging technologies we found interest in RFID technology and wireless communication, together with ontology based data mining

and model based software design. The IT and research field interested by our project is thus the development of complete solutions (integrating hardware and software) deploying RFID technology. The goal is to achieve the skills needed for the integration of ad-hoc IT solutions, a field that at present days is showing the best success chances on the national and international IT market.

1.3 The idea and case study

From a very general point of view, our idea consists in the development of an IS able to automate, through the use of RFID technology, all the procedures involved in the management of those structures sharing the following features:

1. One or more controlled access points;
2. One or more information/help desks;
3. One or more selling points (bars, boutiques, ..);
4. One or more activities taking place in the structure;
5. Valuable assets that could be useful to track (sport goods, furniture, ..).

In order to obtain a more concrete requirements elicitation and analysis, our team got the support by Mondello Immobiliare Italo Belga SA, a firm engaged in the administration of the beach of Mondello (Palermo), that in the last few years is converting part of its business towards the management of beach resorts.

1.4 Approach to the project

The project has been faced following a development model widely used and supported which divides the project itself in several phases:

- Feasibility study
- Project planning
- Requirements elicitation
- Requirements analysis
- Solution Design
- Implementation
- Testing
- Installation
- Project closure
- Maintenance

1.5 Document structure

This paper is divided into three separate parts that have a purpose of supplementing each other in a logical way and of providing a complete final result that has to be consistent and coherent. First part is dedicated to the introduction of RFID technology and all the issues related to it, using existing research and publications; it gives also an overview of the main application fields that can be improved by using this technology stressing on the field relating our thesis project. Second part is dedicated to the path that brought us to develop the present system. It starts from the idea of it (chapter 3), then continues to the application case study details (chapter 4), how does it fit in the idea expressed before and how could it be realized. Following chapters are dedicated to the effective development: from requirement elicitation and analysis of the case study environment, we got a complete document which allowed us to design and develop the system. Chapter 5 will thus illustrate the outcomes of the requirement analysis and design phases. Chapter 6 is dedicated to implementation details and personal contributions. Third and final part is where all the findings and the result from previous two parts come together to form conclusions and answers to the research questions. The system is analyzed in its real working environment and some interesting data is collected (chapter 7). After having conclusions in place, also thanks to the results discovered, thesis ends with chapter 8, dedicated to the project closure, final considerations and future development of the system.

Chapter 2

State of the art of RFID Systems

In the previous chapter we introduced subjects and intentions of this project; now, starting from this basis, we can move on and understand the setting and the environment for its development. For this and any other project to be valid, one of main requirements is to relate it to the existing knowledge and thoughts, in order to create something that it's using all the correct standards and best practices. For this purpose we created a chapter which will introduce the readers to the environment and to the literature we've used in order to succeed in the development of our system. Subjects that will be provided in the following paragraphs include defining and explaining the RFID technology, with all the aspects related to it, the main application where this technology is used and privacy and security issues.

2.1 RFID: what, when, why

In the last decades, Bar code became the dominant standard as a technology for identifying objects. Despite of the simplicity and the low cost feature of it, there were many theorists and practitioners who realized that had to turn into something different as there were great limitations to its use and further development. These people were looking for a new technology that would allow to solve all the limitations that have been afflicting the use of the Bar code. The solution to this research was solved in an amazingly simple way: radio waves. Using radio waves was in many ways superior to what Bar code was able to provide to its users. Good abilities of radio waves and its attributes were well known so they had numerous applications like: radio broadcasting, wireless telegraphy, telephone transmission, television, radar, navigational systems, and space communication.



Figure 2.1. Examples of Bar code

Even though it was clear that radio waves could increase the potentiality of Bar code by maintaining the same principles, the major obstacles were price and undeveloped technology: In fact the technology, in the beginning, was not developed enough to allow feasible application. Roots of RFID go as far as 1940's and 50's when the principle that RFID is based on, was first used in radar technology to identify enemy from friendly crafts. Further on, the development of integrated circuits in the 60's pushed RFID prospects forward. But it was not until Charles Walton pioneered his radio frequency identification technology in the 1970's and 1980's that the real history of radio frequency identification began. With his patents Walton is considered by many to be the father of RFID, he created first electronic door keys that used RFID technology, as we can see from the abstract of its patent[1]:

”‘An automatic identification system wherein a portable identifier, preferably shaped like a credit card, incorporates an oscillator and encoder so as to generate a programmable pulse position-modulated signal in the radio frequency range for identification of the user. The identifier can be made to generate the identification signal constantly or can be made for stimulated transmission responsive to an interrogation signal. The identification signal can be preset or can be programmable by use of a programmable memory.”’

RFID had been around in various forms for years before Walton's invention of a radio-operated door lock. Earlier inventors received patents on animal control systems, a luggage handling system and a mail-sorting system [2]. But Walton came up with a design that is popular today. His technology of the time was good enough and even better than Bar code, but his 1,75 US dollars solution was no match for 25-cent Bar code. In spite of apparent problems, based on his ideas and thanks to great developments in electronics and chip making industries the future of RFID and its use looked bright.



Figure 2.2. One of the first RFID prototypes

Thanks to great development in RFID technology itself and cost reduction, RFID is coming out of the shadows of Bar code and promises to make wonders, if it is to believe many articles, books and research papers that are written about it in recent years. Many,

big and small companies, are working on pilot projects or have already implemented RFID in their business systems. In the past price and technological development were main obstacles and discussion efforts for both developers and practitioners. Today we are witnessing the focus shift from those efforts towards standardization. In order to exploit full benefits of RFID it is necessary to agree on global standards that will allow all parties involved to synchronize business systems, as we'll examine more deeply during this chapter.

Features	Barcode	RFID
Memory	Limited capacity	High capacity
Data access	Static informations, readonly	Rewritable (possibly)
Reading modes	Single reading	Multiple readings
Reading capability	Some cm	From some cm to hundred of metres
Robustness of device	Easily warnable	Resistant when inserted in packages
Interaction	Perpendicular to the label	All directions
Quality	Possible lack of operation due to damages in some supports	Resistant even to dust and water
Price	Economic	From some cents to some euro, still too expensive
Diffusion	Spread all over the world	Still not well spread
Availability	Easy	Low number of productors
Standard	Stable since lot of years	Still evolving
Product packaging	Easy	Not always easy

Table 2.1. Differences between RFID and Bar code technologies

2.2 The technology

RFID is not a single technology, but it's made up of a wide range of technologies very different among each others. This characteristic is due both to structural features and to the differences in the application fields. As a first instance, they depend on the working frequency and on how the tag is electrically supplied. Working frequencies have been decided according to precise international standards, and they cover the range of middle frequencies (125/134 KHz), of high frequencies (13,56 MHz), of very high frequencies (868/915MHz) and of microwaves (2.45-5.8 GHz). As a consequence of the frequency used in a certain context, the reading range will change significantly and consequently the application fields of RFID technology.

2.2.1 The RFID system

An RFID system is made up of three main components:

- the tag (which is also defined with the term *transponder*)
- a reader
- the software system which uses the above components (in some simple application it may be also hardware)

The transponder is applied to the object to identify, and typically stores an identification code, but it can also store some data: for this reason it represents the primary data support of the system. A transponder can be considered basically as a passive support that doesn't own any intelligence and that can be read in a certain instance range (we'll see that this is not always true). The reader is a device that takes informations from the transponder thanks to its antenna that has the ability to communicate with the tag throughout radio waves, allowing reading and writing operations. The software system (it can be a single PC, a host system or in the case of hardware a PLC¹ whose aim is to read the code and to perform some action ².) filters the data read, and makes it available for the informative system.

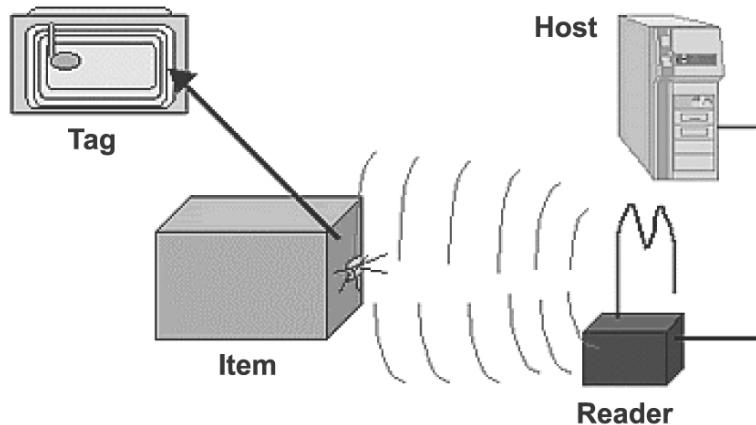


Figure 2.3. A simple RFID system

¹A PLC (i.e. Programmable Logic Controller) is a device that is used to replace the necessary sequential relay circuits for machine control. The PLC works by looking at its inputs and depending upon their state, turning on/off its outputs. The user enters a program, usually via software, that gives the desired results.

²This is the typical case of RFID doors or lockets. These systems are embedded often in electronic devices, and don't need any elaborating CPU to work. The only RFID door feature is to read a tag and decide to open the door or not depending on its ID or on the informations contained in the tag

The TAG (or transponder)

Every item that has to be identified in an RFID system, must have an associated tag: it is the component that applied to an item (an object, animal or person) allows its identification without any direct connection or contact, by sending a signal eventually in response of a command received by the reader. The structure of a tag is composed by the following parts:

- **Memory:** can be, depending on type and complexity of the device, read only (ROM), random access (RAM) or programmable and not volatile. ROM memory is generally used to memorize the tag's identification. RAM memory is useful in order to memorize temporary data during the reading procedure and it saves time. Not volatile memory can be of various types, and the most common is EEPROM³. It's used to memorize tag's data, and must be necessarily not volatile to make sure that data are not lost when transponder is not in reading mode.
- **Finite state machine for the handling of the tag's behavior:** it's needed to handle the information between the tag and the reader according to the standards.
- **Antenna:** it's used for the data communication between the reader and the tag. It must be designed depending on the type of tag and the desired reading instance.
- **Inlay:** it's the structure that connects and protects the components of a tag. It is very important depending on the application field where the tag will be used: in some case a tag must be for example resistant to water, to dust or to high temperature.

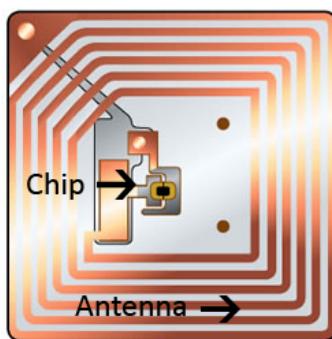


Figure 2.4. Structure of a RFID tag

Tags classification First of all, tags can be classified depending on the energy supply type in:

³EEPROM (electrically erasable programmable read-only memory) is user-modifiable read-only memory (ROM) that can be erased and reprogrammed (written to) repeatedly through the application of higher than normal electrical voltage. Unlike EPROM chips, EEPROMs do not need to be removed from the computer to be modified.

- **Passive:** this kind of tags doesn't contain any battery inside. They take the energy directly from the signal coming from the reader and, since they don't contain any transmitter, they just beam, modulating, the signal transmitted from the reader reflected from its own antenna. They're the most wide spread category of tag, the smallest and the cheapest.
- **Semi-active:** they contain a little power supply, used to give energy to the integrated circuit but also to let the RAM memory work. When transmitting data, they work exactly as passive tags.
- **active:** they contain a battery which is independent from the reader and they have the capacity of transmitting data without being interrogated by the reader. This feature allows the tag to have a better signal, that can also reach hundred of meters. They contains a RAM memory as the semi-active ones.

Therefore, tags can be classified also according to the writing method that they support:

- **Read-only transponder:** the only functionality of these kind of tags is to transmit the UID (Unique identification number) when requested by the reader. This number is associated to the tag during the building process and cannot therefore be modified lately by the user. This makes each tag unique: every object must have a different ID to avoid collisions. UID is made by a sequence of BIT for the identification and control. When a tag is activated by a reading request, it starts to send continuously its UID: communication is based on a single direction (from TAG to reader) and it's not possible any opposite data communication. These kind of tags contain a small and not volatile memory and for this reason they're easy to develop and cheap.
- **Read-Write transponder:** with them it's possible to write data multiple times (except the ones with WORM memory, which allow to write data only once)⁴. Communication is therefore bi-directional even though it's still the reader that continues to interrogate the TAG. The typical capacity for these kind of transponders is about a few Mb for the active ones and some Kb for the passive ones. Usually EEPROM memory is used⁵.

According to these features, tags are divided into different classes, as shown in the table below.

⁴WORM (Write Once Read Many): allows the user to write only once into the memory without the possibility to modify this data. This is useful to modify the TAG's content by itself and not by the productor

⁵User has the possibility to program only one part of the chip's memory: next to a read-only memory, which contains the UID inserted by the productor, the TAG contains a section that can be used by the user and can be modified many times

Class	Transponder type
Class 0	Passive read-only tags.
Class I	Passive write-once tags
Class II	Passive tags with extra features (extra memory, cryptography)
Class III	Semi-active tags
Class IV	Active tags
Class V	Readers: can supply energy for class 0,I,II,III tags and communicate with class IV and V tags

Table 2.2. Classes of tags

The reader

The reader is the element that in an RFID system allows to gather the informations contained into the tag. For this reason it's to be considered one of the key element of an RFID system: its role is to connect the physical world and the application's one, handling the functionality of an interpreter. The reader contains therefore two interfaces, an input one (towards the antennas) and an output one (towards an elaboration center, hardware or software) whose output is typically handled by a middleware⁶. The reader, according to the design and to the technology used, can be a read-only or a read-write device. The main structure of a reader is made up of 2 fundamental blocks:

- **A control unit:** it's the responsible to communicate with the application software, to execute commands, to control the communication with the transponder and to code and decode the signal.
- **A HF interface:** it's made up of a transmitter and a receiver, and it's main function are:
 - Generate the signal to the operative frequency to activate the power supply of the transponder (for passive and semi-active tags)
 - Modulate the signal to the operative frequency in order to send the correct data to the transponder
 - De-modulate the signal to the operative frequency received by the transponder

The HF interface contains two separate paths for the signals, corresponding to the two directions of the data flow from and towards the transponder. Data transmitted

⁶A middleware or integration layer is that architecture layer, software or hardware between devices (like RFID readers but it can be any device like printers...) and the informative systems existent. Its main role is to translate and adjust the raw data coming from the devices into coherent and understandable informations



Figure 2.5. The desktop RFID reader used in the Box Office module of our system

towards the transponder go on the transmitter arm, and data received are processed in the receiver arm.

The antenna Antenna's function is to generate the radio waves in order to gather the response from the tags present into the range of action. Each reader can handle simultaneously one or more antennas. Reader's antennas are bigger than the tag's ones and own protections from external agents. Almost every reader contains an antenna inside, but they may also use external antennas which are connected directly through a cable. Fundamentally, reader's antennas can be of two different types:

- **Mobile:** generally integrated into the readers. They're used manually by the operators or inserted into moving objects; in any case, they're used in cases where it must be the antenna to move towards the tag to identify
- **Fixed:** they can be used in different configurations but they cannot move because they're fixed; in this case it must be the tag to come into the reading range to be identified.

2.2.2 Frequencies and standards

We'll analyze now the different operative frequencies and the standards that uses these frequencies. As we can see, both the operative frequency and the use of passive or active tags affect in a sensible way the working behavior of the RFID system. Thanks to this

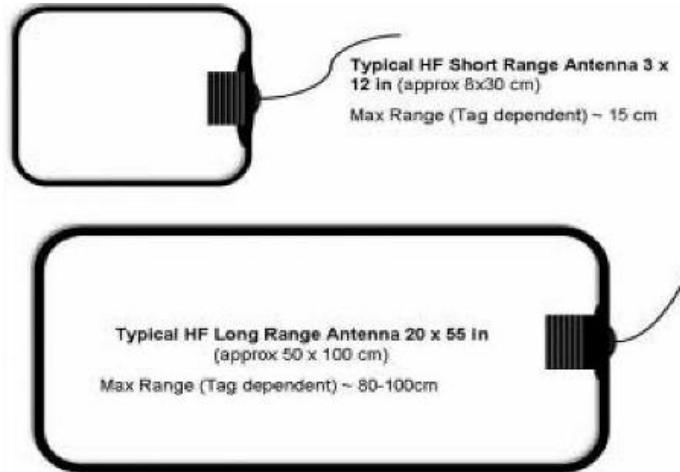


Figure 2.6. Examples external antennas

feature it's possible to use RFID as a common technology for completely different fields of applications.

Frequencies

One of the fundamental characteristics of an RFID system, as we said above, is the operative frequency, that is to say the reader's transmission frequency. Communication frequencies between the reader and the tag depends both on the type of tag used and on the application used and can be different from one country to another. The international and international standardization organizations (in Europe the ETSI⁷ are working in order to realize international standards and solve the limitation deriving from the different choices of each country.) In order to transfer data between the space separating the reader from the transponder, signal is modulated on a carrier signal whose frequency is approximately between 100 KHz and 5.8 GHz. Among this interval, we can recognize the classification explained below:

- **Low frequencies LF (125-135 KHz):** transponders, almost always passive and cheap, receive power supply from the reader and are characterized by having a short reading range. Maximum instance between the reader and the TAG is about 80 cm. The RFID system that use this frequency are used mostly for the accesses control, for the security system and for the animals identification.
- **High frequency HF (10-15 MHz):** transponders, as for the low frequency's ones, receive power supply by the readers. The main features are: short operative range (about one meter), low sensitivity to liquids, high sensitivity to metals, and good speed on write/read operations. RFID systems using this frequency are used

⁷European Telecommunications Standard Institute

mainly for the access control and the most used frequency is 13.56 MHz (worldwide standard). This is the frequency that we used in developing our system.

- **Ultra high frequency UHF (850-950 MHz):** transponders are electromagnetically coupled with the readers. UHF systems are characterized by: high operative range, high speed of read and write operations, high sensitivity to liquids and metals. UHF RFID tags are used mainly in systems for the transportable platforms and containers or for the traceability of trucks. Unfortunately a common standard is still not present and it differs from one nation to another.
- **Microwaves (2.45-5.8 GHz):** transponders are electromagnetically coupled with the readers. Microwaves systems are characterized by: very high operative range and a very problematic issue known as "standing wave nulls"⁸. An example of a system using this frequency is the automatic highway pass (i.e. Telepass), that works on a 5.8 GHz frequency and can solve the standing wave nulls problem because is used on moving vehicles.

We can conclude by saying that an increase on frequency increase also the transmission data speed between the reader and the transponder, but it decrease the transmission capacity in presence of obstacles like metals or liquids.

Standards

RFID itself is "only" a group of technologies: the real innovation will happen only throughout the deeply revision of processes (logistics, payment systems...). As we all know, the innovation of a new technology is directly connected to the standards that are created for that technology: without standards Internet wouldn't have developed as we see it nowadays. Only with the definition of a series of standards it will be possible to see a real development of RFID systems: till when most of the system will be "closed", and not able to communicate among each others because of a not common language, RFID will just be described as a next-to-success technology. There are many standardization entities that are involved in the process of defining technological and application standards for the RFID in order to ensure a correspondence between performances and interoperability of the systems. The difficulty in this point is due to the fact that, as we said before, RFID is not only a single technology, but it's made up of different technologies supporting each others. Besides, the usage of different frequencies is directly connected to the development of RFID for applications of different fields. Some of these applications, for example the ones that uses the proximity cards for the access control, for the transport ticketing or for the animals identification are fully standardized⁹. This has made possible a wide spread

⁸Standing wave nulls: it's a problem that causes the presence of dead zones in the reading range where the tag is not readable. It happens because of the short wavelength of the microwave radiation (from 12 to 30 cm). This affect the system in a way that the microwave system cannot guarantee that the tag is readable if it's not moving, because it could be present in a dead zone. That's why the problem is not present in automatic highway pass systems

⁹See respectively ISO 15693, Calypso, and ISO 14223

offer of applications for these fields. Other fields, like object identification and logistics are still not linked to a well defined standard, and this has slowed down the spreading on RFID based solutions for these applications. For this field, two organizations have proposed their standards, that has some characteristics in common and some basic differences. These organizations are the ISO and the EPCGlobal, whose details are described below.

The ISO Standard ISO (the International Organization for Standardization)¹⁰, which is the world organization for standardization, is strongly involved in the definition of RFID standards. ISO is made up of a big number of committees that are dedicated to the standardization of a specific area. RFID systems depends directly on the area regarding Information Technology whose referent is JTC1¹¹. We'll now list the main RFID ISO standards already existing.

- SC17 Integrated Circuit(s) Cards
 - ISO 7816 IC Cards with contacts
 - ISO 10536 Close coupling cards
 - ISO 14443 Proximity Cards
 - ISO 15693 Vicinity Cards
 - ISO 10373 Test methods
- SC19 Animal Identification
 - ISO 117984/5 Code Structure and Technical Concept ISO 14223 Advanced Transponders
- SC31 Item Management
 - ISO 10374 Freight containers
 - ISO 15960 Application requirements transaction message profiles
 - ISO 15961 For item management-data objects
 - ISO 15962 RFID for item management, data notation
 - ISO 15963 Unique Identification of RFTag registration Authority to Manage the Uniqueness
 - ISO 18000 Air interface standards

In the next chapter the standards ISO 14443 and iso 15563 will be analyzed in a detailed way as they're directly connected to the purpose of our project.

¹⁰For more informations visit: www.iso.org

¹¹JTC: Joint Technical Committee.

Electronic Product Code (EPC) EPC project (Electronic Product Code) was born in 2000 thanks to the studies of a research group of M.I.T.¹² called Auto-ID Center, with the contribution of some of the most important companies producing RFID components. During 2003 the project had an extremely important development: the EPC company was bought by the EAN International and UCC (Uniform code council) and a new company, the EPCCglobal was created. This made EPC as the RFID system that EAN world propose as the standard for the transport of informations. EPC standard defines not only the way to identify object, but also the networks technologies useful to guarantee the traceability of these products during the supply chain.

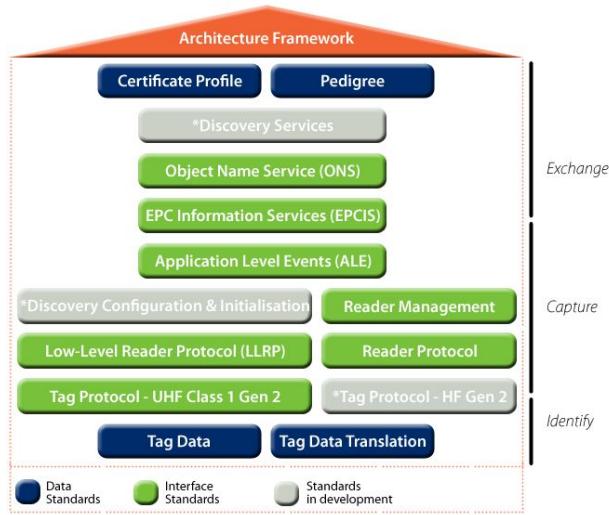


Figure 2.7. The EPCCglobal standards overview

A complete documentation about this project is available at their website. “The EPCCglobal RFID Implementation Cook Book is targeted towards readers who already know about RFID and intend to get involved in using the technology. The Cook Book includes information from EPCCglobal working groups, of which many members are early RFID adopters. The objective of the Cook Book is to provide companies with practical and timely information needed as they prepare for RFID pilot and implementation projects by taking advantage of the learning already captured by the EPCCglobal member community.”¹³

2.3 Typical applications

There are several application scenarios that are interesting to examine. Some of the applications are already well developed and spread in several companies; other applications are just made as a research field. A method for the classification of different application

¹²Massachusetts institute of technology, Boston

¹³The whole book is available at <http://www.epccglobalinc.org/what/cookbook/>

is given by the European Commission, that has created a space freely available where these informations are spread¹⁴. The proposed approach takes as classification element the sensibility level of the application itself. With this approach the European Commission wants to focus the attention on those critical aspects (such as privacy or security) that have been an obstacle for the diffusion of the RFID technology. In order to evaluate the possible security policies to use, it's necessary to analyze the features that determinate the level of sensitivity of the system. These features can be divided as described below:

- **Closed or opened systems.** In this classification a RFID system is categorized on a different way depending on the type, closed (used in a closed environment from a restricted group of users) or opened (when the RFID system is connected to the internet and can communicate with other systems).
- **Always active RFID.** In this classification is evaluated the possibility of having RFID tags always active during all the path of the supply chain.
- **Object to identify.** This classification divides the applications on the object that has to be identified: item, animal or person. In the last case, the tag could be attached to the person itself or to some object belonging to the person in order to let its identification.

We'll now give an overview on the most used application fields for RFID technology.

2.3.1 Animal identification

RFID based animals identification systems in Italy are created in order to help the automatic handling of the breeding activities: feeding, registry, milking. In this field the tags used are low frequency ones applied for example into the animal's ears. The great advantage in the use of RFID for this aim is that all the informations about the animal are directly contained within the tags (therefore the animal) itself, and thanks to this it's possible to retrieve all the informations about the animal in every moment with the only help of a reader, solving the previous system which referred to a remote database, with all the problems related to the network area.

2.3.2 Logistics

One of the main problem that affects the logistic field such as border management, port and aeroport zones, is the good's identifications that transit on these places, that would allow to have an internal and external traceability, and the production of a transport documentation which results to be extremely useful for a better service. When using the traditional identification technologies it's not possible to implement tracking policies with reasonable prices. This limits the companies to apply only the traceability of groups of goods and not of details. Thanks to RFID, there are nowadays systems that allow identification of all the products inserted inside a pallet, allowing therefore the identification of

¹⁴<http://www.RFIdconsultation.eu>



Figure 2.8. Animal with RFID tag

the single product, since each of them contains an RFID tag. The pallet is disposed on a ruler which passes through a relevation door, with a RFID reader, that allows to execute a real time reading of the objects contained inside the pallet and to generate a database with the correct informations. The importance of RFID in this case is vital, since it allows also the identification of products that are hidden inside a bigger box; this feature creates also the problem of parallel reading, since a pallet containing many RFID tag has to be read in a short time: for this reason the use of different anticollision algorithms is preferred and it makes the reading operation faster and more precise. A new and very important application can be found in airport. RFID technology can be used both in the handling of passengers and baggage. An example is in the Hong Kong airport: ”Unlike bar code-only tags, which require the scanner to be in the line-of-sight to read the tag, the AD-833 RFID tag, specifically designed around aviation needs, can be read at long instances without line-of-sight in any orientation. RFID systems are also more reliable, achieving average read-rates of more than 97 percent, much better than a Bar code-only system’s average of 80 percent. As a result, RFID tags allow more accurate tracking of baggage through HKIA’s¹⁵ baggage system. By empowering timely and accurate checked bag deliveries to the planes, airlines and passengers experience more on-time departures. In addition, the system reduces misdirected bags resulting in improved customer service.”¹⁶

¹⁵Hong Kong International Airport

¹⁶<http://www.rfida.com/2009/05/airport-rfid-services-baggage-handling.htm>



Figure 2.9. Baggages with RFID TAGs passing through a gate-reader

2.3.3 Access control

RFID can be considered as a valid alternative to the personal identification traditional technologies (such as badge) and to the strong authentication technologies based on the recognition of biometric features of people. RFID technologies, in fact, doesn't require the visual contact for the identification and allow a instance recognition. This kind of identification makes easier the use of turnstiles and electronic doors, allows to distinguish the entrances from the exit and to verify the list of people inside a structure in a completely automatic way. Besides, RFID technology allows the automatic start of a working session, making possible, for example, the start or shutdown of a pc depending on the person that it's near to it. Tags used in these kind of applications can be printed or inserted in objects of different shapes, like a badge, that can be also personalized with the add of a picture, of logos, text and so on.



Figure 2.10. Example of RFID smart card

2.4 Security and privacy issues

From what we have said above, one could easily conclude that RFID must be the perfect technology, that will solve all our problems and create supply chain and retail paradise. Concluding something like that would be very wrong and one of the most important reasons why such a belief would quickly dissipate is security of RFID. There are two issues that occupy attention of most researchers and business people as well. On one hand there is a genuine concern with personal privacy invasion that RFID will potentially create and on the other hand there is equally worrying problem with data security. In this paragraph we will be dealing with both problems.

2.4.1 Privacy issues

Privacy is more and more under threat from all kind of sources today. We have increasing numbers of cameras monitoring every our move, mobile phones eavesdropping, e-mail control and all kind of computer monitoring and surveillance. These and many other ways of intruding on our privacy are leading to the feeling that we are not alone and safe even in our own places. RFID and especially EPCGlobal's uniquely identifiable products concern and worry many people who believe that this technology will just add more pressure on individual privacy. Before continuing further into this topic lets explain where, how and why this technology affects privacy. Convenient as it is for sellers to know at any moment what they have in their stores, and where it is, the products capability to be traced and tracked in time and space unfortunately doesn't end when the item is sold. So what may happen is that any casual buyer, walking home with bag full of items that can be identified, checked or traced with a simple RFID reader is not even aware of it. On the first glance that doesn't seem very important but considered that if somebody can know what people buy or have in their house, without even coming in, it doesn't sound so harmless anymore: a malicious man can check what do people have inside their houses before planning if to come inside and steal what's inside. This was just an example and there are many more, but what connects them all together and makes them possible is RFID. What we haven't mentioned yet and what is very important in this whole story is that RFID tags must remain active when they leave the store. In order to be read, tags need to remain active and actually deactivating (killing) them is a good way to defend against these and other unwanted scenarios. Because of that, tag deactivation at the exit of the store is likely to become a standard for the whole industry. In cases where deactivation of the tag is not an option, like a new RFID passports that EU countries strive for, shielding the tag (passport) with a metal casing is an alternative option because any metal obstacles between tag and the reader greatly reduce their functionality or disables it completely.

Another option would be to use tag blockers like RSA® Blocker Tags which are "spamming" any reader that attempts to scan tags without the right authorization and in that way keeping the data on the tag secured and privacy guarantied.

2.4.2 Security issues

RFID's potential wide spread implementation in systems like authorized access, toll payment, retail, supply chains and many others, will bring attempts to misuse and tamper with technology. These attempted attacks may range from very innocent just playing with the technology ones, to cases of corporate espionage, forgery and theft. According to RFID Journal¹⁷ there are three primary issues surrounding RFID and the need to protect proprietary information:

- Protecting data stored on the tag;
- Protecting the integrity of the tag (and thus the product);
- Securing data related to the serial number on a tag, which may be stored in a network database.

It was important for our purpose to examine these features in order and to find a trade off between security implementation and costs to apply it. We'll now analyse them in more detail.

Protecting data stored on a tag

When dealing with data stored on a tag there are three main situations that need to be considered: the first is preventing someone from reading the tag, the second is preventing someone from changing the data on the tag and the third is preventing someone from eavesdropping on the communication between the tags and readers.

Preventing unwanted tags reading Even though it might seem that unauthorized reading of RFID tags is not a "big deal" it can actually present a great problem. Reading a single tag from a product basically presents very little danger, the only thing that trespasser can obtain is ISO or EPC serial number of that product and maybe some information about the product (e.g. country of origin, expiration date, etc.). This amount of information seems insignificant and for it is hard to imagine any situation where this information could be misused. The point is that if the data recorded in a TAG contains useful informations about the supply chain, for example, it may be used from another company to improve its process. The same technology that makes it easy to keep track of products and view real time inventories is also making those things possible for others. So if tags are not protected, competitors or anybody else with enough capabilities can keep track of supply chain. All that data can provide very useful facts about business: where raw materials are bought, where they're shipped and much more. Information like that in wrong hands can be very harmful for the unsuspected victim. A solution to this is given by the technique of recording nothing but the TAG serial into the transponder, and to associate each ID with an item in a protected database, or to store data in an encrypted mode. Both of these solutions provoke overhead, but as we just noticed results to be necessary to avoid unpleasant situations.

¹⁷RFID, Privacy and Corporate Data. Available form: <http://www.rfidjournal.com>

Preventing unauthorized data change in a tag There are ways and methods that allow unauthorized change of the data on the tag, in case of rewritable TAGs: this makes data very vulnerable if it is not protected by some sort of software or mechanical protection. Consequences of this kind of shortcoming can be very dangerous. Situation that could then happen is that thief could change the identity of the product, making for example a very expensive item read as a cheap one at the automated cash register at the exit of some store. Another dangerous scenario can be seen in a pallet full of single products whose tags data can be replaced to change the content of the pallet itself. This second case is more critical than the first one for a main reason: in that case, as we saw before, it's more understandable the use of rewritable TAGs in order to keep track of the supply chain for example; in the first case of a single product sold at a store it's more common the use of a read-only tag, whose content cannot therefore be changed.

Preventing eavesdropping on the communication between tags and reader The third issue, in tag data protection sphere, is preventing someone from eavesdropping on the communication between the tags and readers. The problem here is that if someone can tap in on the communication, he/she can gain valuable information about the tag, data on it, encryption methods and/or the reader. All that info can later be used for unauthorized access to the tag, identity theft (substituting identity codes on products), tampering with the data or some other forms of malicious wrongdoing. One of the solutions to this shortcoming, which is suggested by Auto-ID Centre, is to limit the number of times when reader broadcasts an ID number to the tag. The reader can, for example, break up the serial number and only broadcast a part of it. The tags that have the same part of the serial number respond, but this time with the full number. This way full number is only sent in the reverse channel, which is a whisper, compared to the forward channel, which is by necessity a lot louder. By doing this the probability that an eavesdropper will obtain complete serial number is reduced significantly. Another solution is to encrypt communication reader-to-tag and the other way round.

Protecting the integrity of the TAG

When all considerations about data on the tag and access to it are settled, the next step is to think about tag integrity. Physical protection of both goods and systems predecessors of RFID was a problem that companies had to deal with for years, even before RFID. Even though this has been a problem for so many years, it still inflicts great damage on most supply chains. Possible bad case scenarios with this issue are physical removal/damaging of the tag and "killing" of the tag. The Auto-ID Center's Class 0 and Class 1 specifications include a kill command that enables retailers to permanently deactivate the tag at a consumer's request. Some authors believe that the kill command has a serious drawback: it creates a backdoor that those with ill intent could use to circumvent the system. If it is possible to kill the tag in the packaging, then one might be able to put another tag on, with a fake serial number and get a highpriced item for a lower price, but RFID technology itself is much more complex and harder to clone or fake than bar codes so it should improve integrity of data and products. Unauthorized killing can be prevented by

having secret kill code that reader needs to send to the tag in order to kill it.

Securing data related to the serial number on a tag, which may be stored in a network database

One of the solutions to a problem of unauthorized data reading from the tag, as mentioned before, is to keep the information related to the product not on the tag it self but in some sort of database. This database is then connected through Internet or intranet to the readers or other devices that need access to it. Data that is stored in this database can range from usual product specifications to encryption algorithms or keys used on specific tags, "killing" codes and other. In systems that Auto-ID Center specifies to use together with their EPC technology, databases with network capabilities have crucial part and thus these systems are much more vulnerable to this problem than systems with less critical database like most other systems are. Most problems that could occur in this perspective are very familiar to IT experts because they are dealing with them regularly in network-databases environment that doesn't necessarily need to be RFID related. The most highlighted problem is that off unauthorized access to the data in the database. So it is crucial to determine who can access the data and when, which is determined by the company and then it is up to IT technology and experts to keep everybody else away.

Part II

From the idea to the development

Chapter 3

The idea

3.1 Introduction

As already mentioned in the previous chapter, lots of devices with different characteristics exist. RFID is not limited to one specific application, nor it is defined by one specific technology. Radio Frequency Identification technology is a tool, an enabler of functionalities; it is not a goal in itself. Only requirements of specific applications can determine what kind of data will be needed, at what range the tags should be scanned, how the data will be protected and whether there are any concerns like privacy, interoperability and spectrum interference.

RFID technology is rapidly becoming a critical tool for turning company data into valuable business information. The definition of standards and protocols contributed to costs decrease pushing many firms in investing on this technology. In the last 3 years, lots of applications were developed for supporting management and changing and improving companies' processes in healthcare, office automation, transports and educational fields. After many years of experimentation and risky investments, the market seems ready to accept RFID as an advantageous technology on what investing. Therefore, considering all these elements, application developers could take maximum advantage of creating software solutions, capable of catching all its potentials analyzing companies' needs.

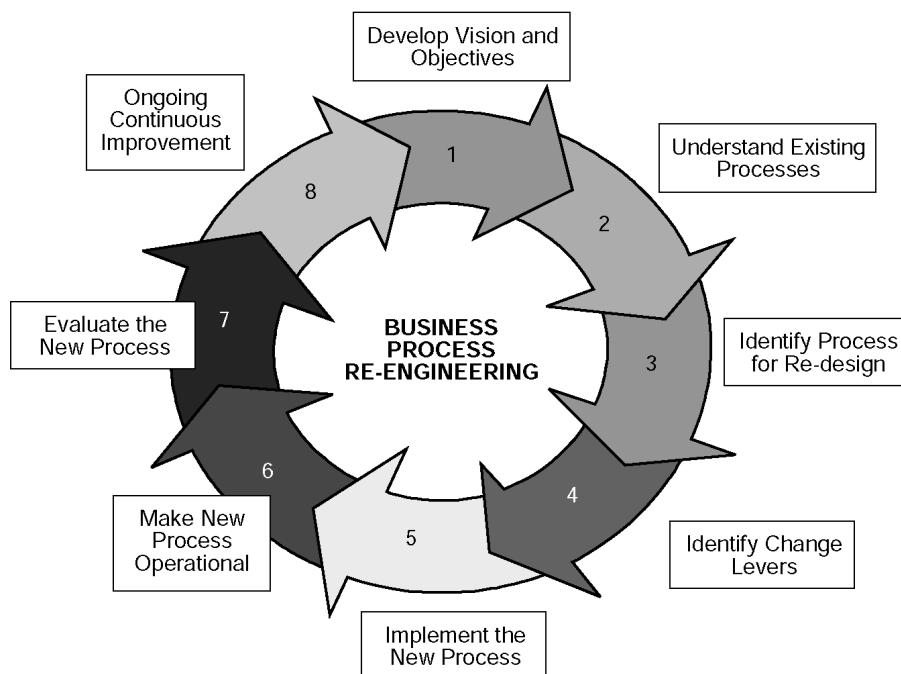
The purpose of this chapter is to identify RFID potentials and analyze the market situation, in order to delineate the major features of a generic software suited to provide a valid support to any kind of leisure time and touristic structures through radio-frequency identification of guests/customers.

3.2 Features of leisure time and touristic structures

In this paragraph we'll briefly analyze common processes-activities of some commercial and entertainment companies that share common features, in order to identify the major functionalities of our software solution idea, describe them and understand how an IS (Informative System) RFID-based can result advantageous.

3.2.1 Role of the ICT and BPR in Corporates

The introduction of ICT in companies open the way to new processes. Technology is not a simple support for organizations, but becomes a way of changing. It introduces new opportunities, otherwise impossible to persecute without its presence. The overall activities by what a firm reviews its processes, in order to improve efficiency and efficacy indicators, can be grouped with the acronym BPR (Business Process Reengineering). A necessary review of their own processes has to be done to better answer to the rapid and continuous market changes. The BPR pushes to new forms of organizations based on the usage of the technology in order to increase productivity, quality and offer new services. To maintain a high level of competitiveness, the company has to prevent changes, rapidly percept market needs and immediately face them. All these actions can be accomplished only by using a technology infrastructure and an Informative System that can support the management decisions and can help the continuous reengineering of the company's processes. In figure 3.1 we report the flow of the major activities to be conducted to reengineer company's processes



Source: Vakola *et al.* (1998)

Figure 3.1. Business Reengineering Process

3.2.2 Introduction to a specific application context: touristic resorts, restaurants, parks, sports centers

A brief analysis on RFID potentials and features convinced our study group to try to understand how deploying this technology to develop an application to leisure time and touristic structures. These are commercial establishments, used for relaxation or recreation, attracting visitors for holidays or vacation. Ideated for satisfying vacationers' needs and wants, they provide customer food, drinks, lodging, meetings, sports, entertainment and shopping services. Examples these kinds of establishments are: sports centers, touristic resorts, exhibition fairs and amusement parks. In most of these structures, customer's access is usually managed by a check-in procedure that typically takes place in a reception-desk/box-office, located at the entrance, in charge of an entrance fee payment or services selling (available furniture in case of a beach resort, available room in case of hotels and villages, available space in case of any exhibition fair or amusement park). After check-in procedure completion, customer should present to an access control point (e.g. an automatic gate or an employee), equipped with a proof of his access right. Passed the gate, he/she could be interested in getting some information about the activities and services respectively taking place and offered in the structure; therefore, a help desk can satisfy customer's needs, guiding him in moving all around the village-fair-park. The presence of activity points of interest or services and goods selling points will finally provide him food, beverage and any other facility to be enjoyed.

An example of a touristic resort map and its major features is shown in figure 3.2.



Figure 3.2. Example of a Resort Map

We can therefore list all the common features shared by the previous mentioned structures on what we'll focus after, to make our idea a concrete software solution.

A sports centre, an exhibition fair, an amusement park or a touristic resort is commonly characterized by:

- **One or more Check-in desk:** it is intended as the entry point of the structure, by what customer acquires access rights for it. While some structures need to enable the customer's access only by a fee payment (case of amusement parks, single entrance in sports centers), others have also to register his credentials(resorts, fairs).
- **One or more controlled access points:** after a fee payment and/or a registration, the access to the structure has to be hardly controlled to avoid the access of unauthorized people. This aspect becomes critical for that structures whose only way of making profit is the payment of an entrance fee.
- **One or more information/help desks:** spread all over the structure, they help customer in moving inside it and enjoy the services offered.
- **One or more selling points** (bars, boutiques, ..);
- **One or more activities taking place in the structure:** especially for that ones who admit a limited number of participants (sports centers, resorts) a booking procedure is needed.

Considering the previous front office activities (activities visible to the customer), a company can also be interested in:

- **Managing valuable assets (sport goods, furniture, ..):** a constant tracking of structure assets helps operative processes to be performed easily, and the management in minimizing investment costs.
- **Registering every commercial transaction.**
- **Registering customer details and personal interests:** knowing their interests and major info, corporate could focus more on their customer needs and offer specific and personalized services.
- **Allocating and coordinating efficiently its human resources in order to minimize resource allocation costs, particularly influent on company's balance.**
- **A periodic monitoring of the structure:** checking customers and assets to avoid unauthorized access and services fruition to guarantee a maximum security level.

3.3 OUR idea

3.3.1 Features

Considering the major features briefly treated in the previous paragraph, our idea is: realizing a system capable of accompanying and facilitating the customer in enjoying all the things he is interested in, from the first access to the fruition of the services, from the food and beverage purchasing to the benefiting of any other facility inside the "Village". A real important aspect, is making clear the real philosophy message under the idea: customer should feel part of a community, whose components are joined from the usage of a special RFID tag equipped "accessory" (can be a particular and fashionable wristband or a jewel) that makes customer free from the usage of bills inside the "Village", and that allows him/her to receive a quick and personalized service. The RFID market offers lots of possibilities about the choosing of the accessory's shape and material in what integrating the RFID tag, giving the highest flexibility to a management that persecutes any kind of marketing actions and would capture customers' interest.

3.3.2 The RFID tag equipped accessory

The first innovative aspect that customer will find in his own hands, will be the fashionable accessory chosen as the case for the RFID tag. Thanks to modern production techniques, this accessory can be made as completely water and weather proof, and used therefore in many open air contexts like parks and villages. Its design should exalt the corporate's image proposing a highly attractive style also capable of winning customer's resistance to change. Modern production techniques, let corporate choosing over lots of pre-defined shapes and materials (as shown in figure 3.3), or to design the RFID tag case that better suites their own needs.



Figure 3.3. Examples of wearable RFID tags

Considering the wireless techniques used for the communication, particular attention

should be posed over the customer's health problem, considering he is continuously in strict contact with the RFID tag accessory, since he/she needs to wear it/keep it inside the structure. These accessories' operative frequency defined by "proximity" and "vicinity" ISO standards (suitable standards for applications involving people), is absolutely lower than any other common device's operative frequency, like mobile phones or Wi-Fi access points, so they cannot cause any scientifically proved damage to customer's health.

3.4 Discovering and analysis of application context requirements

After briefly described the major functionalities characterizing structures like touristic resorts, parks and fairs, let's focus our attention on that elements trying to discover their major executive problems keeping in mind ICT potentials. Problems identification will then guide us in deciding how to intervene for solving them by developing an IS deploying RFID technologies.

3.4.1 Check-in desk procedure

Description

All activities related to customer's registration and access are usually taken at a reception desk located at the entrance of the venue. After **checking the access availability** (different depending the specific structure), a customer could provide his personal details and pay for a single access (parks, fairs), a subscription (sports centers, beach resorts) or an accommodation (hotel resorts) to access it. A **proof of his access rights** will then allow him to pass through the access control points spread inside.

Activities analysis

Let's decompose the check-in procedure to deeply analyze the activities it implies.

1. **Checking of access availability.** Allowing customers' access implies a precise check of the structure's resources availability. While an hotel and a touristic village should check accommodations' availability for a certain period, an exhibition or fun fair should have to be sure that there is space to host new customers without overcrowding.
 - **Problems:** An old manner way of checking accommodations and space is registering customers' reservations and tracking entrances on paper modules. Although it seems a practical solution since everything was managed in this way before the introduction and growth of ICTs, in medium-large companies this information management process would require a lot of time and resources spending. Basically, for a hotel room reservation for a time period ranging the instants t_1 and t_2 , the booking office should check that, between the all overlapping reservations, there is at least one room available for that period. This can

be done only by consulting and analyzing every booking module ranging the period considered, after, certainly, have rescued them from a big archive. While for a BandB structure, that owns just few rooms, it couldn't be a time and resources-consuming task, for big hotels and resorts it doesn't result feasible. A consistent reengineering action has to be executed.

- **Changing Opportunities:** Introducing a software system, capable of creating and managing accommodation reservation or customers' accesses, registering when necessary also their credentials, will allow company managing easily, quickly and without any human mistake, this principal task. A booking office can now easily evaluate a room reservation just querying an electronic Database in a few seconds. The dimension of the archive is strictly confined to a normal PC, and the consultation time is reduced for more than 95 percent compared to what previously needed.
- 2. **Access control points:** releasing a proof of payment. After a brief checking for access availability(available furniture in case of a beach resort, available room in case of hotels and villages, available space in case of any exhibition fair or amusement park), the customer could normally pay to enter and enjoy services/accomodations offered by the venue. A proof for his/her acquired access rights should be provided to let him/her demonstrating at the access control points(e.g. employee equipped) he is allowed to enter in.
- **Problems:** Paper or plastic pass/card are commonly used for this task. Their big problem is that they are simple to clone having few cheap machines and a low graphic design know-how. This results in a loss of money, particularly influent for that structures whose most part of profit is related to an entrance fee payment. Particular attention should be posed on how the access control points are realized. The usage of people could not guarantee a strict control procedure, since in many contexts they let enter unauthorized people because pushed by working interests or kinship relations. A change is required.
- **Changing Opportunities:** RFID tags and automatic access control devices(tripods or gates) will perfectly face the previous problems. Thanks to EPCGlobal serial code definition, RFID tags are uniquely identified by a UID set during the manufacturing process by the producer. This code cannot be cloned since only the producer can imprint it on the chip. In addition, tripods or gates RFID reader equipped offer management a precise access control activity and the possibility of registering every customers' admission. Management could therefore benefit of this data, discovering hot access time intervals in order to allocate and coordinate efficiently its resources.

3.4.2 Information/help desk points

Description

After the check-in procedure completion, customers can easily enter in. Here, being new to this context, he/she could need some help in knowing features and points of interest located inside. Usually located in strategic points (e.g. near the entrance, in the middle, next to other principal points of interest) are help/information desks, designed to offer customers help and advices to make their stay pleasant and comfortable. A customer can simply wait his/her turn and ask employee what he/she wishes to know.

Problems Especially in big structures, capable of hosting a huge number of people in their big extension area, many help desks are needed. Their realization implies also a big number of employees, subtracting human resources to other more important operative tasks. Therefore, these high personnel costs introduced by the help desk realization, bring companies to spread some of them all over the structure. This could cause, especially in crowded periods, queues and the related inconveniences to customers' stay.

Changing opportunities In order to avoid long queues at information/help desks, it would be desirable to provide the structure with one or more automatic informative kiosk-totem (equipped with an RFID reader), offering an easy interface designed to guide the customer to the information he/she needs. They could be also capable of executing, lightening therefore its workload, some of box office activities like customers' info modifying and activities subscription. Assuming to use RFID tag like a debit card for an electronic payment system(see next paragraph), kiosk could also offer some credit account management functionalities.

Synthesizing, depending also on the application domain, it should be possible to:

- visualize (and in some circumstances modify) the profile associated to the RFID tag read by the kiosk;
- subscribe to scheduled generic activities;
- get information about scheduled activities;
- make a deposit on the credit account associated to the RFID tag read by the kiosk, and visualize its status and customer's transactions;

Management could in this way employ human resources in other operative activities, charging kiosks of executing more static and schematic tasks(tasks that don't require human strength or particular attention), like the information supplying and management one is. The activities subscription and credit account management functionalities will instead decrease box office workload.

3.4.3 Goods and services selling points

Description

They are intended to be the points of interest charged of selling food, beverages and/or generic services to satisfy customers' major needs and wants. Once inside customers can easily purchase bar products, rent sport furniture and enjoy any other facility related to every specific application domain.

Problems A continuous monitoring of customers' transactions is needed. Not rare are the cases in what employees, in charge of get payments, cheat, subtracting money avoiding a fiscal registration of a payment. In many open-air structures, like summer villages or aquatic parks, customers usually wear beach clothes like costumes or light dresses, not always suitable to keep a wallet or simply bills and coins. It results in a consistent decrease of customers' purchases, from what, clearly, many of the commercial structures derive the most part of the profit.

Changing opportunities Using the RFID tag equipped "accessory" as a debit card to implement an electronic payment system, allows us to perfectly face the two problems arisen in the previous subparagraph. Each selling point should be equipped with all the necessary (e.g. PC, software module and RFID reader) to allow the sale of goods and services by using only the RFID tag. Cases of employees' cheats will disappear since no payment can be done without it, and every transaction will therefore registered in database, becoming useful also for a successive reports consultation. Particular attention has to be posed on security issues related to electronic payment. Avoiding to register credit account data on the RFID tag memory, and protecting every transaction with a password insertion, are only two of the most principal countermeasures to be applied. Collecting customer purchases information will also allow management to analyze customers and structure principal trends.

3.4.4 Monitoring structure activities

Description

A periodic monitoring of the structure is needed in order to avoid unauthorized access and security problems. Usually equipped with wireless devices(e.g. walkie talks), employees charged of checking structure's security are spread all over the area. They could be used by management to avoid unauthorized accesses, to check if a customer has the right to receive a service(e.g. activity subscription) or to use a generic equipment (e.g. furniture rental), that valuable assets are in the right position and in general controlling the fluid proceeding of everything happens inside.

Problems Checking customer's rights requires the employee of owning instantly information about every structure's transaction. While in little-dimension structures, this monitoring could be managed by a simple employee eyesight, since he/she can remember

what customers accessed and what they did during the visit, in big structures this way to proceed is not feasible. Employees in charge of monitoring cannot complain with the customers if they just "think" that a customer is not allowed to have access or to enjoy a service, since they don't have correct and complete information to demonstrate this. Real time information providing is needed.

Changing opportunities After computerized check-in desk procedure, help desk and selling points, in order to register instantly every structure's transaction, corporate could additionally provide monitoring employees an handheld device such as a PDA or similar (equipped with an RFID reader), wireless connected to a Database server, in order to have the real time situation of the structure. The employee will now be able, instantly, to check:

- if a customer owns the appropriate access rights (e.g. if he/she can access to a particular area)
- if a customer paid to enjoy a service (furniture rental, activity subscription, bar products etc.)
- customer's profile information, after obtained his/her consensus, for security reasons or for accomplishing additional goods and services sale tasks(e.g. selling of goods and services by PDA).

... simply, only by his/her RFID tag "accessory" reading.

The analysis completed above, allowed us to identify all the features and related potential opportunities given by the deployment of every single module to automate the common processes and activities shared by structures like touristic resorts, exhibition fairs and sports centers. Since the objective of this work is to aggregate our academic competences in order to face a great challenge, developing a concrete and eventual commercial software solution, our study cannot exclude, at least, a brief competitors analysis in order to discover the state of art of similar software applications already proposed. Therefore, in the next paragraph we'll conduct a quick research on similar software solutions in order to refine and align our software requirements and functionalities to the present market situation.

3.5 Market competitors

While the born of an innovative idea is usually inducted by casualty and not from economic or competitive reasons, its realization, in a concrete product/solution, is the result of a long production process that implies high money-time investments and risks. The market history is characterized by many excellent ideas that couldn't be, for many reasons, converted into profitable projects. Looking only at idea potentials and promises without considering also economic and technical external-internal aspects would bring corporate to a unsuccessful project. It's for these reasons that a firm, group of developers in our case, before realizing any project idea has to:

- Analyze external situation, identifying threats and opportunities coming from a particular market sector-segment
- Analyzing internal situation, identifying firm's weaknesses and strengths related to internal processes and competencies
- Make strategic and operative choices, in answer to previous analysis for evaluating successively related results.

These are the principal activities that characterize the "Porter-Mullin model"; a model that underlines the importance for a corporate to continuously analyze the market situation, customers, potential entrants, substitutes and suppliers, in order to react to them and maintain a high level of competitiveness.

A deep market analysis following Porter's model could require a large amount of time, should be conducted by experts of market analysis field and it's out of the scope of this paper. On the other hand, we chose to conduct a brief analysis of market competitors that could result advantageous for understanding, defining and increasing our software solution features and value.

3.5.1 The research

The objective of this research is to briefly analyze Italian competitors' solutions in order to identify missing functionalities and understanding how offering a more complete solution increasing, eventually, our level of competitiveness. Due to low time and resources availability, we'll conduct only a "secondary research" characterized of gathering information through indirect sources like websites, news, publications, advertisement. It is a quick and economic research type, that can be conducted by everyone considering also it implies low precision and high level knowledge of the market situation. In the next paragraphs we'll list the competitors we identified during our research on software solutions for managing touristic structures like resorts, villages and sports centers, focusing for each of them on their solutions' features and functionalities, identifying consequently threats and opportunities for our system.

3.5.2 F.C. Consulting Group: SummerMagic

FC Consulting Group S.P.A. is a team of professionals, managers and consultants specializing in management consulting, training management, communication and technology services associated with them. They developed different software applications deploying RFID technology for ticketing, monitoring and access controlling. In march 2007 they released a software to manage and automate some villages and beach resorts processes and activities called "Summer Magic". "Summer Magic", says one of the developers, "thanks to RFID technology, has made it possible to optimize the control procedures through the use of special cards and plastic bracelets equipped with a waterproof radio frequency tags that, in addition to the normal recognition of user data, lets realizing an electronic payment system".

It's a program capable of:

- managing check-in procedure, creation and management of customers' subscriptions
- executing access controls, continuous checking and logging of customers' accesses using automatic RFID gates.
- executing electronic payments(developed only in 2009), RFID tags are also used like debit cards implementing therefore an electronic payment system similar to the credit cards one.
- populating database with structure information, it allows to populate DB with all information needed(e.g. customers' category, services and subscriptions details) to allow the previous functionalities working correctly.

How emerged from some web articles, they conducted an experimentation collaborating with few beach resorts located along Ostia beach in 2007, running only check-in procedure and access control functionalities. By an our phone call to their offices, in October 2008, we discovered that they planned only for the season 2009 to have the electronic payment system implemented. It seems there are no other articles on Seasons 2008 and 2009 software deployments and this would make us thinking about their limited influence on this market segment while it's reasonable they are continuing working in Ostia surroundings.

RFID technology used

They use RFID cards working at the operative frequency of 13,56 MHZ, characterized by a RW memory of 2KB dimension. These tags are suitable for applications involving people identification since the frequency used is quite lower than the ones used in mobile applications.

Threats and opportunities

Compared to the functionalities we planned to introduce in our software modules, Summer magic is a few functionalities and simple software. It doesn't cover important activities like structure monitoring (we propose instead PDA module) and help desk ones(we propose Informative Kiosk emplacement). They just propose to use RFID tags like magnetic cards or Bar codes, like an entrance pass, forgetting another important feature like the RFID tag-based payment system that was maybe developed just in 2009, after 2 years of activity. They could base their strengths only on the fact they are in this market segment since 2007 but their software appear really meager. The usage of RFID cards characterized by a memory dimension of 2KB let us thinking they store customers' credentials or generic information on RFID tag; while we'll be going to store all customers' data only on a protected server(guaranteing strict privacy policies) using wearable RFID tags(e.g. wristbands). The GUI, based on the Microsoft Access masks, appears heavy, old and poor, not aligned to the windows XP or Vista advanced style.

3.5.3 WSC - World System Consulting: Quick Suite

WSC - World System Consulting has been on the market since 1998. It is defined as an "Information and Communication Technology corporate specialized in touristic sector and with significant experience in the automotive one." WSC has put at the heart of its solutions RFID technology, maturing a 360° experience thanks to the many projects that have been developed in different business contexts. It owns the headquarter in Turin with offices in Sydney and in Sardinia. It's an IBM partner since 2003, and Microsoft and Cisco certified since 2004. They developed different software applications deploying RFID technology for electronic payments, automotive, logistic, ticketing and access controlling all grouped under the name "'Quick suite'". In 1998 they were involved in "Valtur villages" processes reengineering in order to automate/computerize some of their principal tasks and activities. Reviewed commercial and financial areas at first, they started to improve those ones strictly connected to customers. They therefore realized their first module, that is now grouped in the Quick suite, whose name is "'QuickPay'". After few years(2003) they introduced "'QuickEntry'" and "'QuickRIGEL'" enlarging their Quick suite functionalities. These modules are ideated to be simply composed to form a complete software solution according to committee needs and wants. For structures like resorts and villages they propose a composition of "'QuickPay'", "'QuickEntry'" and "'QuickRIGEL'".

This system is capable of:

- executing electronic payments through "'QuickPay'" module, RFID tags are used like debit cards implementing therefore an electronic payment system.
- executing access controls through "'QuickEntry'" module, continuous checking and logging of customers' accesses using automatic RFID gates; additional biometric devices can be requested.
- managing bar selling points through "'QuickRIGEL'" module, allowing to manage bar goods orders ,lounge tables and register all the payments. Additional RIGEL PDA modules enable to make mobile orders quickly and without any usage of paper. QuickPay module intercation is used to execute RFID tag payments.

Visiting their press review, emerges they are continuing their collaboration in reengineering and computerizing "'Valtur Villages'" processes intervening in numerous different areas:

- contability
- fiscalization
- knowledge management
- human resources management
- customers check-in procedure

For march 2009 they planned to revolutionize Village resort of le Flamboyant (Mauritius). They will allow customers in using an RFID card(quick pay) for executing electronic payments and accessing their rooms. The new introduction of "Quick domotic module" will allow management to see instantly when the customer is out of the room in order to execute efficiently all facilities management operations. It seems they reached a good reputation already in 2004 appearing in famous newspapers "Il sole 24 ore" and "La Repubblica", and other tourism magazines and websites . In these articles we find references only to "Valtur Villages". WSC can be considered therefore a big corporate since "Valtur" owns about 25 villages spread between Europe and Africa.

RFID technology used

They use RFID cards working at the operative frequency of 13,56 MHZ, characterized by a RW memory of 1KB dimension. These tags are suitable for applications involving people identification since the frequency used is quite lower than the ones used in mobile applications.

Threats and opportunities

As already underlined for Summer Magic, WSC offers a solution with few functionalities that cannot assume a strategic importance for their customers. They base their commercial campaign only on Quick Pay advantages; many advertisement articles were found on this module but nothing else on the others, and nowadays they cannot pretend to be innovative only proposing a relatively (they introduce in 1998) new way of payment. RFID cards could result uncomfortable in villages and beach resorts where people wear light dresses or beach costumes, that are not clearly suitable to keep this QuickPay card. Instead we propose wearable RFID tags like wristbands or necklaces. "Valtur" appears as their only client in touristic field. This lets us thinking there could be possibilities to eventually commercialize our software to other structures.

Chapter 4

Application case study

4.1 Introduction

The scope of our idea got clear and sharp through the considerations exposed in the previous chapter. But keeping this level of abstraction makes it impossible to proceed to requirements elicitation and analysis. Furthermore, the overview on market actual situation provided at the end of chapter 3 suggests B2B (*Business to Business*) as a better choice compared to B2C (*Business to Consumer*) in the context under consideration. The next step is thus clear: seeking for the chance to interact with an existing firm to better focus on domain specific issues and opportunities. The firm Mondello Immobiliare Italo Belga SA (from now on referred as MIIB in the rest of this document) has a century long history and has spent the last decades managing a 2 kilometers long beach thanks to a concession given by the Commune of Palermo. This collaboration with MIIB gave us the right input to leave the level of abstraction evident in the previous chapters of this paper to focus instead on a clear case study: a beach resort. Throughout this chapter we'll illustrate:

- why the beach resort managed by MIIB is a suitable case study and effectively an instance of the more general commercial/entertainment structure presented in chapter 3 (paragraph 4.2).
- the way we intended to proceed on reengineering the processes affected by the introduction of an IS deploying RFID technology (paragraph 4.3). This paragraph is further divided into sub-paragraphs to explain how we intended to proceed to each step of our method;

4.2 Focusing on the beach resort

First of all it's important to show how a beach resort has all the features presented in chapter 3 making it interesting to approach this kind of touristic structure with our idea in mind. To do so, it could be helpful to recall these features following the same order selected for the more general case:

- One or more **Check-in desk**: everybody who wants to enter a beach resort has to pass through a check in desk to pay the entrance fee and/or other kind of money amount related to the fruition of the services offered by the beach resort.
- One or more **controlled access points**: most of beach resorts have a flexible gate control procedure since guests should be able to exit and enter the structure by demonstrating their right to enter each time. The Italian law is also clear about the obligation that beach resorts leave a margin of at least 5 meters from the sea shore that has to be available to the public and a public access from the street to the sea every 150 meters (reference to the sicilian regional law: D.D.G. n. 476 of the 1 june 2007 REGIONE SICILIANA Assessorato Territorio ed Ambiente). Thus, even in a more strict access control model, it's impossible to impede to a guest to reach the nearest access to the street and pretend to enter the beach resort again from the main entrance. Since the check-in desk would be overwhelmed by the access control procedure, it's necessary to think to one or more controlled access points.
- One or more **information/help desks**: whenever a beach resort pretends to offer more services to their guests aside to the only beach furniture (i.e. activities, courses, music), the need for information will automatically arise. This is why an information/help desk would be needed.
- One or more **selling points** (bars, boutiques, ..);
- One or more **activities** taking place in the structure: as mentioned before, a big number of domain related activities can be organized to assure a pleasant and active stay to the guest inside a beach resort.

4.3 Our approach to process reengineering

4.3.1 Project hypothesis and Feasibility Study

As summarized in the previous paragraph, a beach resort is a perfect application for an IS deploying RFID technology. To better understand the interest moving MIIB to make such a collaboration with our study group, it's certainly useful to depict the last managerial attitude of the company, making more clear how an hypothesis of a BPR phase results helpful to their business. Though this company is been engaged for such a long period into beach management issues, until few years ago their business model was restricted to:

”Building and placing a certain number of small wooden bungalows on the beach, sold before the season start and for all the season period, limiting then the access to the beach to the only clients showing a access card with the bungalow number on it at some access points”

We'll refer to this model as the **Bungalow Model**. In the last years, after a significant change in the management board, the company is living a period of deep revolution due

to a concrete shift in the business model. As the number of seasonal bungalow placed on the beach is decreasing, the space left free is addressed to an increasing number of independent beach resorts offering single beach locations consisting of a beach umbrella and of a certain number of beach beds (according to the number of clients per location). This model is widely spread in the rest of the Italian coastline and the MIIB is progressively following the trend even if with a consistent delay. While this position gives to the MIIB the chance to accurately consider pitfalls and benefits deriving from the choices made by their competitors, the lack of experience in such a managerial approach is evident as company's processes are not naturally compliant with the new business needs.

From a very general point of view, this new model offers a bigger daily mean profit but exposes the company to a new broad range of managerial issues and variables. The services offered to the client are different in features and number compared to the Bungalow Model. Even the number of beach furniture has increased with a consequent need for a more efficient control over their usage and allocation.

The hypothesis of a project in conjunction with MIIB became more concrete after some considerations concerning how such an innovative step would fit in the process of renovation taking place in the last few years inside the company. RFID technology, though widely tested, is not quite spread outside the fields of logistics, transportation and animal control (at least in Italy at present days). This is why the innovative component of its application in such a context doesn't have to be underestimated. Along with this point, the RFID tag case chosen has to be as stylish as possible.

No company would ever start an innovative project without evaluating costs and benefits first. The most spread mean for making such evaluation is a feasibility study. The goal of this document is thus to provide company's management board with the set of information necessary to decide for a real kick off of such an IT project and for the relative investment. These information concern the technical and operative feasibility, the benefits, costs, risks and deadlines intended for the development of an IS deploying RFID technology: Mondello ITBe@ch.

The feasibility study

The feasibility study we prepared is divided into 5 sections:

The actual situation In this first section of the document the scope of the study has been defined. The strategic view of the company was reviewed in terms of services, organization and technology.

Later in the section, problems and opportunities related to the introduction of an IS were listed; they were organized into categories: existing problems noticed during the study, general opportunities catchable even without the choice for the introduction of the proposed solution and opportunities arising in the scope of the project comprising those

opportunities that could derive from the introduction of an IS. The information have been organized in a table that we propose here.

Existing problems	General opportunities	Opportunities arising in the scope of the project
Lack of control at access points		Access control through RFID enabled turnstiles.
	Access monitoring to collect information on presence trends inside the structure.	Precise and fine-grained information collection and statistics customized on management needs regarding structure attendance.
Lack of a precise control on location management.		Introduction of an IT procedure to allocate locations to clients enabling a location level control.
	Set clients free from cash need inside the structure.	Providing clients with an electronic payment system based on a wearable tool.
	Services and goods' purchases monitoring in order to activate a customer fidelity program.	Precise and fine-grained information collection and statistics customized on management needs regarding customers' purchases inside the structure.
	Selling ads spaces to be shown to customers inside the structure.	Introducing an intelligent system of ads rotation on one or more screens spread throughout the structure based on the kind of customers inside the structure at every moment.

Figure 4.1. Problems and opportunities for the ITBe@ch system

Then the actual equipment of ICTs at MIIB was presented along with the identification of project's constraints. These constraints were concerning the Italian legislation, schedule and other kind of context specific constraints. Concluding this section, a paragraph was dedicated to project's goals. These goals resulted to be:

- Strict access control
- Strict control on valuable assets
- Making cash unnecessary inside the structure
- Supporting strategic decisions
- Minimizing impact on staff and customers
- Improving MIIB's image on the market
- Respect of economic and timing constraints

Solution overview This section of the feasibility study was dedicated to solution's requirements in terms of:

- Affected procedures
- Operative methods
- Architectural requirements
- Make or Buy choices

Risk assessment In the feasibility study document, a high level risk assessment was presented. This was further enlarged and improved during the planning phase. This is why we postpone the description of the Risk Analysis phase later in this chapter.

Proposed project To better appreciate each one of the solution's features, the proposed project was divided into modules.

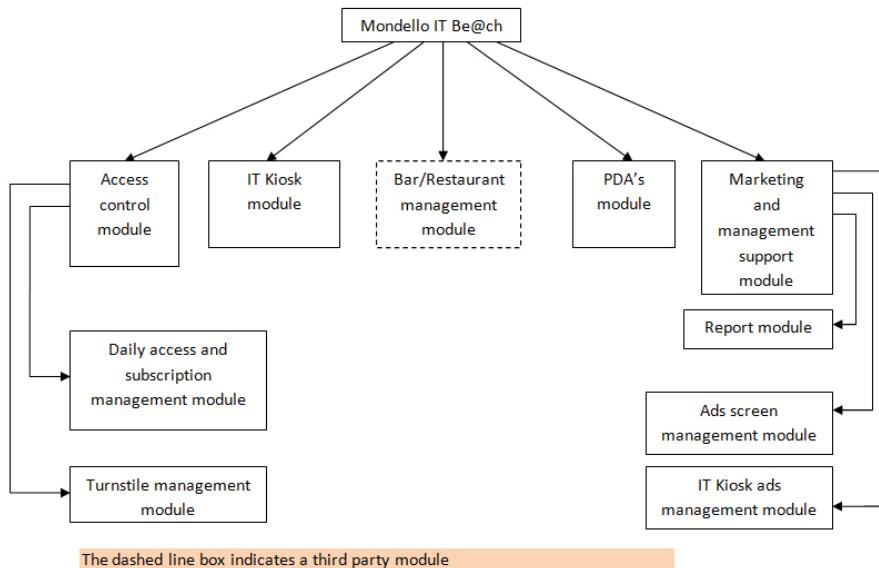


Figure 4.2. System modules

Then the project has been segmented into subsequent phases without a specific timing contextualization (relegated to the planning phase). We conducted a high level planning of the project that enabled us to make an hypothesis on the sequence of deliverables (both documental and applicative) and the checkpoints where MIIB management board would have been able to evaluate project's outcomes.

Benefits/Costs analysis The analysis we performed lacked of a certain level of detail reachable only with more solid skills in economic evaluation tasks. Benefits identification refers first of all to cashable benefits, that is benefits corresponding to a measurable decrease in actual business costs or an increase in business profits. Second of all, we presented those benefits corresponding to the modification of an observable and quantifiable phenomenon (most of the times regarding time or quality aspects).

Cashable benefits

- Expenses due to operative staff working on a beach resort exploiting IT Be@ch should decrease since the introduction of RFID enabled turnstiles should substitute at least one resource per access point;
- The usage of a wearable RFID tag as a mean for electronic payment should increase the revenues for goods and services sold inside the structure since:
 - Several studies confirm an increased number of “spontaneous purchases” related to the avoidance of cash;
 - With a service from the bar directly at the location, it’s easy to imagine a larger tendency to purchase.
- The introduction of an IT kiosk (totem) and of a certain number of flat screens will provide space suitable for advertisement. item The arrangement of an access control system will consent to avoid abuses with a double impact: to set free the location previously occupied by abusers and to increase the number of regular clients.

Non-cashable benefits

- Service quality will increase due to:
 - Easy payment system;
 - Direct selling at locations;
 - Quick access through RFID enabled turnstiles;
 - Improved information delivery to client; he/she will be able to quickly get information regarding activities taking place in the structure, events, announcements and so on, through the interaction with the kiosk.
- Staff skills will increase: the use of an IS will push the staff towards a personal and professional growth.
- The amount of information available to management, useful to make strategic decisions, will increase.
- The amount of paper documents will decrease.
- Operative processes and procedures will have a speed up and an improvement in efficacy.

- The image of MIIB will improve: the use of an innovative system will enlarge the consensus around the company (positive comments, media etc.).
- A reduction in service delivery mean time.

To analyze non-cashable (but measurable) benefits it's necessary to provide a set of metrics:

- Task execution time;
- Service delivery time;
- Amount of information;
- Quality (efficiency, efficacy and client satisfaction)

According to these metrics the phenomenon related to the expected benefit will be measurable. MIIB is not able to provide any data on previous years so the only way to signal an improvement is in a relative and not absolute fashion, based on the typical positive impact of this kind of IT projects. More in detail, talking about the service quality improvement:

- The ease of electronic payment will be measurable through the observation of the number of persons preferring the RFID tag instead of cash.
- Service improvement related to the direct selling at locations will be measurable through the percentage of purchases made through PDAs.
- The larger information fruition by the client will be measurable through the number of visits to the kiosk.

The improvement in staff skills will be measurable through typical parameters widely used by human resources control office. The increase in the amount of information on the beach management will be evident along with the reduction of paper documents. Speed and efficacy of operative procedures will be measured by these metrics:

- Task mean execution time;
- Number of trespasses and irregular fruition of services inside the structure. MIIB's image improvement will be measurable through:
- Number of claims;
- Number of positive comments;
- Number of positive articles or tv citations.

Since there's no data on previous years, clients and staff opinions will be crucial to analyze the situation after IS's introduction.

Costs categories These are the main costs categories:

- Human resources employed to fulfill the project;
- Hardware required by the system;
- Software licenses for the use of third party software;
- Deployment and start up of the system;
- System management costs;
- Assistance and maintenance.

Project kick off and Project Plan

The feasibility study has been presented and approved by MIIB during a board of directors meeting. Next step after a feasibility study that has been approved is the planning phase. The result of this phase is the Project Plan and generally a set of attachments. In our case the document consists of 5 parts as follows:

First part - Introduction This part simply introduces the document and lists document's content

Second part - Project overview The project overview provided in this part of the Project Plan aims to define the problem statement and the general project description. Furthermore a correlation between the outcome expected by stakeholders and enabling project goals is depicted. Then the scope of the project is delimited within a series of boundaries to explain what is and what is not part of the project. Finally some preliminary assumptions that the reader should keep in mind while reading the successive parts of the Project Plan.

Third part - Project Team and stakeholders The project involved some key figures inside the MIIB executive and operative staff. The interfaces between the development team and these figures had to be defined. This is why in this section of the Project Plan the structure of project team is presented along with the connections with stakeholders inside MIIB and the responsibilities of each team member.

Fourth part - Project kick off This is the true heart of the Project plan since this section contains the definition of project's lifecycle, planning approach, methods, tool and techniques to adopt, estimates and estimation techniques, project activities (explained through a Work Breakdown Structure), project schedule, human resources and budget allocation.

Fifth part - Support plans This part provides a brief description of the attachments of the Project Plan. (see table 4.1)

We will now summarize the most important parts of the Project Plan that show how we first imagined to approach the several tasks of such an IT project.

Attachment	Description
Budget management plan	Budget management plan defines procedures, tools and methods necessary for the estimation, allocation, monitoring and control of project's costs. After a first estimation of allocated budget, it is mandatory to employ an operative method to refine the estimation and evaluate the discrepancy with the first estimate.
Communication management plan	This is a plan meant to define procedures, internal and external communication means, information distribution methods and stakeholders management.
Purchases management plan	This document provides indications regarding procedures to adopt in products or resources acquisition.
Quality management plan	This plan illustrates procedures to adopt in definition, analysis and monitoring of project's quality.
Risk management plan	This plan illustrates procedures to adopt in definition, analysis and monitoring of project's risks.
Configuration management plan	This is a document that aims to explain tools and techniques to use in order to keep always a set of stable versions of the developed system and documentation at each project phase. It also defines the approach necessary to make any change to any of project's variables.
Human resources management plan	This plan illustrates procedures to adopt in definition, analysis and monitoring of project's human resources.
Gantt diagram	This diagram graphically illustrates project's scheduling.

Table 4.1. Project plan attachment

Problem statement MIIB wants to employ an Informative System (IS) that:

- Provides a solution to several management issues concerning beach resorts;
- Enables also to take advantage of some missed opportunities;
- Constitutes a supplementary input towards MIIB's image improvement, compliant with the latest strategic choices.

Project's stakeholders are:

- Positive stakeholders, that is the ones that expect a benefit deriving from project fulfillment:
 - MIIB's management;
 - MIIB's shareholders;
 - Beach resorts' clients with medium or good lean to technology usage;
 - Project's team members;
 - Personnel oriented to professional growth and stimulated by technology usage;
 - Suppliers of project's and product's enabling technologies.
- Negative stakeholders, that is the ones that expect a damage deriving from project fulfillment:
 - Competitors
 - Clients with low lean to technology usage;
 - Personnel seeing technology as a threat to their professional and personal skills.

Project overview Project consists in the development of an IS that automates a good percentage of processes forming the management of a beach resort and that in the same time provides a valid support to short, medium and long term strategic decisions. Project is starting from scratch meaning that no technology support is present for any one of the processes affected by system introduction. The development will consist of a part dedicated to software components and a second one concerning the integration of the software obtained with a series of off-the-shelf hardware products (meaning that these hardware components are not internally produced).

Project's results and goals With the feasibility study approval, MIIB has shown its interest in the achievement of several peculiar results. These results can be directly mapped to a identical number of project's goals that should enable their achievement, as the table 4.2 shows clearly.

Results	Goals
MIIB's image improvement	Development of a system deploying an innovative and attractive technology (RFID) that meets all project's positive stakeholders' expectations
Improvement in quality of services provided to clients	Development of a system that increments the level of efficiency, efficacy and user satisfaction for provided services
Automatic and effective access control that prevents abuses	Implementation of an automatic mechanism that couples the access to the structure with the mandatory payment of a ticket.
Permeation of a proprietary electronic payment system inside the beach resort	Development of a system that makes clear to the client the great advantages and ease of use deriving from the adoption of an RFID based electronic payment system
Achievement of a stable information basis for short, medium and long term strategic decisions	Development of a system able to track all events revealing attendance and purchases trends in order to process the collected data providing a summarizing and useful view.
Strict control on beach furniture and services fruition inside the beach resort	Development of a periodic rounds mechanism, supported by technological devices, intended to reveal any abuse inside the structure

Table 4.2. Classes of tags

Project's scope This paragraph has to define the boundaries of the project explaining clearly what has to be considered as part of the project and what is conversely excluded from it. To facilitate the identification of project's scope it's convenient to frame it in terms of final products that the project aims to produce in order to achieve MIIB's business goals.

Project includes:

- Development of the IS Mondello ITBe@ch complying with the requirements presented in RAD

- Installation of the system in the desired location/s
- Personnel training
- User manual
- IS source code
- Project's documentation
- IS documentation
- Assistance

Project excludes:

- Promotional campaign of the innovation
- Sale of system advertisement spaces
- Marketing plan enabling a complete capitalization of system potentials
- Not-programmed modifications to system functionalities after system installation
- Software updates and annual upgrades

Critical Success Factors The analysis of the proposed project brought to the identification of the following Critical Success Factors (CSF):

- Business specific factors
- Ability to attract new clients due to the interest aroused by the use of innovative technology
- Alignment of the project to the process of improvement of Sicilian touristic business quality

Factors related to the strategic positioning of the company

- Ability to enforce MIIB's position compared to known competitors (beach resorts in Capaci, Isola delle Femmine, etc)
- Improvement of MIIB's image

Cultural factors

- Ability to overcome the distrust of clients resistant to change
- Ability to overcome the distrust related to RFID technology as a threat to personal safety, health and privacy

Organizational factors

- Ability to motivate personnel about system's appeal.

Timing factors

- Development and installation of the system before the start of season 2009 (1st May 2009)

Project's lifecycle Along with the activities suggested by literature on Project Management, it is necessary to define the set of activities provided for software development. As a fundamental step in such a project, a lifecycle model has to be chosen in order to identify activities' categories, set the way to execute them and the deliverable they will produce. The term "software lifecycle" refers to the way a development methodology or process model decomposes the activities in correlated sub-activities whose final result is the product itself and all the documentation associated to it. In literature it's possible to find several models for the piloting of software lifecycle. These models define some guidelines to follow in building a system. Model selection depends on project's dimension, degree of knowledge on the application domain, desired degree of flexibility and on the three constraints of the "iron triangle"(budget, time, resources). The most used models are:

- Waterfall
- Incremental
- Evolutionary

The limits imposed by the first two models when there's a high level of initial uncertainty on application domain, pushed us to the adoption of the evolutionary one. This model is useful to reduce defects and misunderstandings deriving from initial uncertainty, since it's characterized by a high level of requirements flexibility in medium projects. It's based on a tight collaboration with the committer, through the release of prototypes whose functionalities are progressively expanded. This is why this model results to be suitable in the examined context.

Planning approach In order to identify the activities to run for each phase of Project Management we decided to follow the model proposed by Project Management Body Of Knowledge (PMBOK®). This model, approved by PMI (Project Management Institute) defines all guidelines for processes, activities and deliverables enclosed in each phase of every project. The adoption of this methodology as a reference for PM processes identification and execution has two major reasons:

- PMI has a long and renowned experience in projects of all sizes;

- The methodology recommended by PMI is widely used by thousands of companies in several business areas.

The Project Plan submission to MIIB's management was programmed as a pass/fail checkpoint for the project. Scheduling and budgeting details of the plan convinced MIIB to pass to the Requirements Analysis phase which involved the preparation of a Requirements Analysis Document (RAD).

Risk assessment Project's risks are to be considered as events that has a negative impact on it. From a more general point of view the term should include all the events that potentially have any kind of impact on project, no matter if positive or negative. According to this last meaning, the risk management activities define the approach and the set of techniques, tools and methodologies to adopt with a double goal:

- To control and mitigate the negative impact of risks within a precise framework;
- To adopt all the measures needed to take full advantage of risks with a positive impact on the project.

The PP was delivered along with a Risk Management Plan (RMP) defining, first of all, the general approach to risk management, that is the set of activities provided for it and the time that has to be reserved for them. Then in the document the first phase of risk management is analyzed in details: to identify, analyze and plan response to risks. The last part of the RMP is dedicated to the explanation of methods and tools to use during the phase of risks monitoring and control.

Risk management approach Project's risks management comes untied into three major phases:

1. A planning phase in which the RMP is produced;
2. A phase for the arrangement of the risk management framework, useful for the kick off of the next phase, to be executed right after the end of the planning phase;
3. A risks monitoring and control phase that covers all the rest of project's duration.

Processes identification and requirements elicitation

The first section of the Requirements Analysis Document has to present the actual situation of the company. This involves a deep understanding of organization processes, internal procedures and policies. Of course the scope of our analysis was limited to those processes that were meant to be affected by the introduction of our informative system. A picture of the company was necessary to achieve a correct requirements elicitation for the solution. Here is a brief description of what we found inside MIIB and the considerations around these features. As mentioned at the start of this chapter, MIIB has a century

long story but the business model adopted until few years ago was quite different from nowadays. In paragraph 4.3.1 we referred to the old model as the Bungalow model. A full set of company's processes have been designed and improved through years to fit that model. The involved business compartments are:

- Executive management
- Sales
- Accounting
- Operative beach staff

The replacement of an increasing number of bungalows with beach resorts is not making the Bungalow model obsolete since a certain percentage of bungalows are going to stay in the future. This is why we'll describe this model as an actual one, aside to the newer one of beach resorts that we'll present later in this paragraph. According to the Bungalow model, executive managers have to set up the price for bungalow's seasonal rental. Before the start of season, typically in February/March, a promotional campaign is conducted asking to interested persons to show up at MIIB's offices during a ten-days-long rental campaign. A priority is given to the renters of the previous year and sometimes no bungalow's left for new potential customers. Before 5 working days from the date of contract subscription the accounting office has to receive a proof of rental payment. Once the payment is confirmed, the rental is complete and the client is provided with 8 access cards each with one of the 8 names provided by him during contract preparation. At the start of the season clients receive bungalows' keys and a staff members positioned at beach accesses will have to control people's possess of the personal access card at each crossing. Now this is surely a simple model to follow. The product offered is sold completely before the start of the season without any need to care about a bad weather day. Nevertheless a diversification in the way the client could make use of the beach was needed for dozens of reasons. A broad range of personnel was completely unprepared to such a revolution and is still showing a hard resinstance to change. Besides transition issues, clients pool is giving a quite good feedback to this strategic shift so MIIB's management is very far from a reverse gear. What is anyway clear is the lack of tested procedures and internal policies concerning beach resorts' management. As we figured out this situation, we quickly found both a negative and a positive aspect: the great level of managerial uncertainty would have affected our requirements elicitation but, at the same time, was giving us a certain level of flexibility for building up a good system for MIIB. MIIB is concretely testing its managerial model against real life situations and issues. The addressing of coastline portions to daily customers, along with other strategic decisions made in the last years, is having a deep impact on MIIB's business processes with the natural consequence of an experimental period evaluating the solutions adopted and to adopt. It's in this context that a vast number of indicators is fundamental to elect a preferential business model. Indicators available only with a detailed collection of information. Let's now briefly examine the procedures adopted before the introduction of our IT solution in the scope of beach resorts.

The Resort Model The business model implemented by MIIB in the field of beach resorts envisages a 5 months long season, from 1st of May till 30th September. It is possible for the client to make use of beach resorts' services from 9 am to 7 pm. An exception has to be done whenever a rest stop or pool hall is present (bar, pubs or similar), which gives to the clients the opportunity to enter till late night, even though the beach locations are not accessible. A typical beach resort has one controlled gate and another one, facing the sea, uncontrolled and most likely uncontrollable. The clients can choose among occasional access on a daily basis and several kinds of periodic subscriptions of different duration. The beach location has a standard furniture with two sun beds, a director's chair and obviously the parasol. No system is present to automate the daily access procedure. The client has to consult the pricelist, ask the box office employee the desired service and, after the payment of a fee, he receives a non fiscal receipt that gives him the right to access the resort and to the paid service/s. At the entrance, another employee receives the client, helping him to orientate himself in finding the right location. No information is asked to the client to help his identification. The clients who want to make a periodic subscription has to pay an amount to make use of a standard location for the requested period. Against the payment, the client receives two access cards with no name on it, but just the location number and the period of the subscription. It is possible to add a third person to the subscription by paying a supplementary fee. Children less than 10 years old enter for free. The smaller period for a subscription is 1 month. The client is asked to give name and surname even though, as mentioned, the access cards are not personal.

Human resources A typical beach resort employs one resource at the box office (check in desk), a door man, and a certain number of beach assistants adequate with the dimensions of the resort. In this context it's not relevant to enumerate the lifeguards, since there would be no interaction between them and an hypothetical IS. In case the resort has a bar with tables, the bartender is helped by waiters/waitresses.

Additional services Until now the only additional service consist of the renting of lockers (dimensions more or less 35x50x40 cm) useful to leave personal belongings. In some cases kids entertainment is provided.

Strategic indications for the future The percentage of sand addressed to beach resort in Mondello is gradually increasing while the bungalow model is conversely decreasing. The actual management, that aims to re-launch the company in many ways, lives a phase of experimentation in order to optimize a business model able to achieve the best results in terms of quality of service, assets usage and personnel optimization. The way MIIB has taken requires some adjustments that has to be considered in the requirements analysis:

- The daily access has to be slightly modified so that clients can choose to enter in the afternoon and pay a smaller fee for the half day.
- Several additional activities could take place in the beach resorts and this let us make the hypothesis that the number of supplementary services will increase soon;

- The type and level of services offered in the several beach resorts managed by MIIB could become different so that different kind of clients could feel attracted by each of them.

One last note has to be made on MIIB's vision on IT equipment and support: MIIB recognizes IT equipment's potentials but in most cases just as a support to business processes. This means that technology is seen mainly as a set of technological tools that, when adapted to preexisting business processes, can automate some routine procedures. This sensation is confirmed by the use of a software of accounting automation, one to support the bungalow renting (both these two software have been realized especially for MIIB) and a standard level of office automation software (Microsoft Word, Excel.).

Processes and requirements analysis

After the description of the actual situation, the RAD continues with the solution proposed. This section aims to present a detailed description of the system that should become a reference both for developers and committer. The goal is then to agree on the functionalities that the system should expose and on how the actual business procedures have to be modified in order to comply with the new system. The description of the solution proposed will pass through a general overview, functional requirements analysis, non-functional requirements analysis and several UML models fundamental for the communication with the committer. The outcome of the requirements analysis phase will be examined deeply in chapter 5. This is why we recommend to the reader to refer to the next chapter to better get the features of the whole informative system.

Solution design

After we found an agreement on system functionalities and solution requirements, we moved on to solution design. The output of this phase is a System Design Document (SDD) that describes accurately all the design choices made by our team. In particular, the document presents design goals, system decomposition in subsystems, hardware-software mapping, data management, access control to each functionality and boundary conditions. In chapter 5, along with the mentioned requirements analysis, the reader will find an overview of the design choices regarding each one of the subsystems forming IT Be@ch.

Solution implementation and testing

Until now we developed and tested a subset of the system so that not all the modules present in the RAD and SDD are ready. The development details provided in chapter 6 are not covering all the functionalities presented in chapter 5. Nevertheless the part we developed is for sure the heart of the system so that the missing modules can be considered as appendices that we'll deliver by time. Chapter 6 will focus on implementation features at a good level of detail (accordingly with the purpose of this documentation) so that, after that the shape of what we wanted to build should be clear, the reader will be able to understand how we actually made it. Though the testing tasks took a lot of our time before

issuing our solution, these were mainly walk-through tests that are difficult to document in this context. An overview of the testing approach will be presented anyway.

Chapter 5

System analysis and design

5.1 Overview

As explained in chapter 4, MIIB needed an IS able to support a wide range of processes and to enable a more structured execution of the several activities gravitating around beach resort management. The situation before the introduction of the IS should be clear (even if not in details due to the space constraint imposed by this document) from the previous chapter. At this point a clear description of the functionalities intended for the considered IS and of the design guidelines has to be provided. Thus the aim of this chapter is to describe both the analysis phase and the design one, in order to illustrate the path that we followed to develop IT Be@ch. As we've identified before the modules that composed the system, we'll first have a look at the system requirements analysis; afterwards, the system design will complete the overview by giving details on how we designed each module. When possible, some contents of this chapter have been extracted from the Requirements Analysis Document and the System Design Document. Those parts will therefore appear as the development part is still to-do.

5.2 Approach to Requirements analysis

The requirements elicited through process analysis and interviews with employees and managers are often in a raw format. The next step is then to analyze and organize them following a set of formal models. One of the most spread approaches is the modeling through UML patterns and diagrams, that's the one we've chosen. Here's a list of the models used in UML 2.0:

- Scenarios
- Use cases
- Use cases diagrams
- Class diagrams

- Sequence diagrams
- State diagrams
- Activity diagrams

Scenarios are a concrete, focused and informal description of a single feature of the system as it would be used by a single user. For our project we produced 40 scenarios according to the template of table ??:

<i>Name of the scenario</i>	<u>ScenarioName</u>
<i>Partecipating actors</i>	<u>ActorInstance:Actor</u>
<i>Events flow</i>	1. Event 1. 2. N. Event N

Figure 5.1. Template of scenarios

A **use case** in software and systems engineering is a description of a system's behavior as it responds to a request that originates from outside of that system. In other words, a use case describes "who" can do "what" with the system in question. The use case technique is used to capture system's behavioral requirements by detailing scenario-driven threads through the functional requirements.[3]

For our project we produced 88 use cases according to the template in fig. 5.2.

<i>Name of use case</i>	<u>UseCaseName</u>
<i>Partecipating actors</i>	Initiated by: InitiatingActor Communicates with: Actor1, Actor2,...., ActorN
<i>Events flow</i>	1. User action 2. System response 3. 4.
<i>Entry conditions</i>	List of conditions to be met in order to the use case to begin
<i>Exit conditions</i>	List of conditions to be met after flow of events completion

Figure 5.2. Use cases template

After use cases are written, a general view of them can be provided through **use case diagrams**, useful to depict the relations among actors and use cases and among use cases themselves. One of the primary uses of **sequence diagrams** is in the transition from requirements expressed as use cases to the next and more formal level of refinement. Use cases are often refined into one or more sequence diagrams. For our project we produced

44 sequence diagrams. 19 **Activity** and 6 **state diagrams** complete the range of dynamic models used in our project. The documents we indicated above were out of the scope of this paper: for this reason, in the following paragraphs we reported the functionalities that were obtained thanks to the development of the UML diagrams. We divided the system into subsystems as already introduced in the previous chapters, and we associated the discovered functionalities to each module.

5.2.1 Emerged functionalities

Box office

The box office is probably the module offering the wider range of functionalities. Aside to the fundamental check in procedures, a set of client management functionalities are necessary. The access to these functionalities is dependent on a login procedure:

Box Office Login: this task is needed to get full access to the system. The procedure asks the user to insert the personal credentials (the credentials chosen are as simple as username and password).

Once the user is logged in, all the functionalities are available. The only exception to be made is relative to those functionalities that need the RFID reader to be connected to the computer. If the RFID reader is not detected by the system, those functionalities have to be disabled. To better enumerate box office's functionalities we propose a clustering in 4 areas:

- Check in area;
- Access to client management area;
- Periodic client management area;
- Daily client management area.

Check in area. This area encloses the functionalities necessary to clients check in (both in a daily or periodic client case).

Daily access: box office's employee has most likely to process a big number of daily customers. The idea is thus to enable the processing of groups of people in a single procedure, allocating clients on a certain number of locations. The fundamental steps regards client's data insertion, tag assignation to each of them and the assignation of a number of locations compliant with the number of customers. Optionally it should be possible to enable a virtual account for one or more clients and assign some supplementary services to locations. At the end of the procedure the bill can be partial, if just a subset of assigned locations is selected, or total. Costs records will comprehend requested location/s and service/s, the bail for tag usage and eventually the amount charged on virtual account/s.

New subscription: as for the previous years, locations renting campaign will start some months before season start. The procedure is thus necessary in company's office preventively to season start. According to observed trends, only few locations remained

available for subscriptions at the box office. Nevertheless the possibility to make a subscription directly at the box office will increase the level of flexibility of the system. The subscription for one location is intended for a maximum of three adults. First of all it will be necessary to set subscription's general features (type, period, eventually a discount). Supposed that at least one location is available with the required characteristics, this location has to be selected and reserved until the procedure is completed or canceled. Passing to the next step, clients' data have to be inserted, category (adult, child, etc.) and interests. As for the daily access, each client has to have a tag assigned and eventually an amount on a personal virtual account accessible through the tag. Optional services added to the location will be provided throughout all subscription's duration. Once the check in procedure is over, the client has to physically cross resort's gate.

Access control: the controlled gate has to be provided with RFID enabled turnstiles and therefore a functionality is needed to consent clients discrimination based on their right to enter.

Access to client management area. The functionalities in this area share the goal of giving to the box office's employee a quick access to a periodic customer profile or to the situation of a daily customer, in order to activate the functionalities intended for the two types of customer. The employee has substantially two choices:

- **Via tag management:** in case client's tag is present, this functionality provides the quickest access to client management section. Once the tag is correctly read, one of the followings could happen:
 1. A periodic customer profile is opened
 2. A daily customer situation is shown
 3. A message signals to the employee that the tag is not assigned to anyone.
- **Via parameters management:** useful to provide an alternative way to clients management, this functionality is needed when client's tag is unavailable. The chance to use this procedure also for the daily customer depends on the quantity of data collected during the check in phase. The functioning in fact needs some client's personal data to be inserted. The system will then execute a search of clients that meet the inserted parameters, providing a result of it. Then the employee should eventually be able to select one of the resulted records and start the visualization of periodic customer's profile or daily customer situation.

Periodic customer management area. Once the employee has got access to the profile of a client, the number of available functionalities is such to require this dedicated area. Here's an overview of the functionalities:

Client's profile modification: client's personal data have to be modifiable after a periodic membership has been subscribed.

Password modification: the password used to authorize payments through virtual account has to be modifiable according to client's preference as long as the format is acceptable (indicatively 5 ciphers like most PINs)

Subscription visualization: If more than one subscription is present for the searched client, his profile should show all of them, giving the chance to select each of them and visualize the relative characteristics.

Services and activities visualization: the set of activities and services bought by the client has to be visualized providing the chance to analyze each of them in details.

Receipt printing for access to activities and services: a right to access to the bought service or activity has to be shown to the relative provider. Thus a functionality is needed to print the receipt when the client needs it.

Subscription termination: the termination of a subscription intervenes when the client brings the tag back to the box office. If the tag is not evidently damaged or lost, the employee will have to give back to the client the bail and eventually the amount remaining on the virtual account. This money restitution should happen through cash or through a coupon that gives to the client the right to get this money on his/her virtual account whenever another subscription or daily access is made. As a note, the choice for the coupon should be motivated by some incentives not in the scope of our study.

Transactions visualization: box office's employee should be able to visualize all the transactions made on the virtual account of a client in terms of date and time, reason and amount.

Tag substitution: when a tag has to be substituted, the employee should be able to signal all the possible cases: tag externally damaged, unreadable tag (though externally undamaged), tag lost, tag readable and undamaged but to be substituted anyway. After the substitution reason has been signaled, the employee should be able to pass to the assignation of a new tag.

Location substitution: For several reasons, the client could want to substitute his/her location. That's why a location substitution procedure should be provided. A substitution should be possible just in case at least one of the locations (available for that period) is compliant with client's subscription.

Virtual credit recharge: in every moment the box office's employee should be able to recharge client's virtual account through cash or coupon.

As for the periodic customer, an area is dedicated to **Daily client management**. Since all the functionalities presented for the periodic customer are similar in case of a daily customer, the repetition will be avoided and only a list will be made:

- Personal data modification
- Password modification
- Services and activities visualization
- Receipt printing for access to activities and services
- Daily customer check out (equivalent to subscription termination)
- Transactions visualization
- Tag substitution

- Location substitution
- Virtual account recharge

The last functionality intended for the box office is needed when the working day is over. **Box office shut down:** when the day is over, the system has to be shut down and the eventual number of tag that didn't make return to the box office signaled to the employee. The employee will have to signal that those tags are lost and the bail will be kept.

Remote interface

The Remote Interface is the module that in a software is commonly called as "Backoffice" or "Backend". Its main function is to fill the database with information that will be necessary for the other "Front-end" modules to work. In other words, it's used by the head of the company (that's why the use of the "Remote") to control the static tables in the database. For this reason the functionalities are mostly defined as CRUD functionalities (Create, Read, Update, Delete) as we'll describe later. This module contains also another very important functionality: the Report analysis. This feature is extremely important for the management of the company: knowing the trends of the customers, the time span when the structure is more full, the less popular services or activities and much more can help the managers to improve the business of the company. As most of the functionalities are similar, we won't describe all of them; we'll list which kind of data will be treated with CRUD functionalities and describe which are the other functionalities added to this module which differs from them.

First of all, let's list the data that this module will interact with:

- Activities
- Advertisements
- Beaches
- Categories of clients
- Discounts
- Employees
- Locations
- Types of locations
- News
- Preferences
- Tags

- Types of payment
- Types of subscriptions

The functionalities related to the handling of this kind of data, as we said above, share a similar structure that is made up of an access to the section, a visualization of the list of the elements of the section, a visualization of one particular element, an insertion of a new element, a modification of an existing element and a deleting (if possible) of an existing element. For this reason it would be a long and unuseful job to repeat the same description for all of them. We'll instead describe the features that characterize every data handled in this module.

Beaches are the main containers of all the other informations. All the data described below refer to a certain beach. For this reason it will be available a section with CRUD operations about the beaches. Besides, it will be possible to associate a map with every beach, and this map can be filled with **locations**. Locations can be managed throughout CRUD operation as well in a visual way, so that the map is WYSIWYG¹, in a way that the user will be able to move locations in the map and to see in real time where the selected location will be placed. Locations can be of different kinds: for this reason, a CRUD section for **types of locations** will be provided. By doing this, different prices and characteristics can be associated to every location in the map.

News and announcements are a mean of communication between the company and the customers. Any important information in the structure (such as new activities or a change in prices) should be placed in this section. This section follows the CRUD structure, even though it contains an additional feature: an announcement which is present in the database, can be published or not. It means that not all the data of this section contained into the database will be visible to the customer. For some reason an announcement doesn't need to be deleted but doesn't have to be showed to the customer anymore; in this case, a functionality of "publish" or "temporary remove" will be provided. The same feature will be available for the **Advertisements** section.

The **Employees** handling is used to give the right access to the future users of the system. Since the company is already using a system for the treat of the data regarding the employees, the only important informations in this section are username and password of the employee. For this aim, a section providing CRUD operations for this will be present in the Remote Interface.

Services and activities that MIIIB wants to make available in the structure must be every time inserted into the database, specifying all the characteristics necessary to allow the selling. In the case of activities, these must also be programmed where necessary in terms of date and time. For this reason also services and activities require CRUD operations, while activities require also programming functionality addes.

Categories of clients consists in the differentiation of the customers according to the parameters choosen by the company's management, in order to offer different prices and services depending on the fact that the customer is an "adult", a "child" or for example a "student". Also in this case is therefore required CRUD operations. The area

¹What You See Is What You Get

dedicated to the handle of "types of subscription" is necessary to offer different types of subscriptions (e.g. monthly, daily and so on). For each type of subscription it will be possible to specify a list of features and the price. This will be handled thanks to the CRUD operations.

Opinions expressed by the customers about the activities and services available inside the structure will be handled in the system, in order to obtain useful informations that can be used in the reports. Also in this case CRUD operations will be provided.

RFID tags which will be used by the customers and by the employees will be also inserted into the database before being used. For this reason in the Remote Interface a section regarding both the "**types of tags**" (to allow in future the use of different tags like necklace or smart cards) and the tags management will be available. For each kind of tags it will be possible to set the value of the bail for its use. Also in this case, CRUD operation will be required, and a functionality of research will be added: it will give the possibility to look for the single tag by reading its RFID ID.

Look, for completeness, we'll give the list of all the functionalities we identified in the present module:

- Login
- Logout
- Activities visualization (list)
- Activity visualization (single)
- Activity insertion
- Activity modification
- Activity deletion
- Activities schedule visualization (list: it consists in a calendar with all the activities planned)
- Activity schedule visualization (single)
- Activity schedule insertion
- Activity schedule modification
- Activity schedule deletion
- Advertisements visualization (list)
- Advertisement visualization (single)
- Advertisement insertion
- Advertisement modification

- Advertisement deletion
- Beaches visualization (list)
- Beach visualization (single)
- Beach insertion
- Beach modification
- Beach deletion
- Categories of clients visualization (list)
- Category of clients visualization (single)
- Category of clients insertion
- Category of clients modification
- Category of clients deletion
- Discounts visualization (list)
- Discount visualization (single)
- Discount insertion
- Discount modification
- Discount deletion
- Employees visualization (list)
- Employee visualization (single)
- Employee insertion
- Employee modification
- Employee deletion
- Locations visualization (it consists in the visualization of a map, where a location can be identified with a jpg image previously saved in the "types of locations" data)
- Location visualization
- Location modification (modification of the data related to a certain location, e.g. number and furniture)
- Location position modification (modification of the position of a location inside the map; it doesn't affect all the data saved about the location which is going to be modified).

- Types of locations visualization (list)
- Type of locations visualization (single)
- Type of locations insertion
- Type of locations modification
- Type of locations deletion
- News visualization (list)
- News visualization (single)
- News insertion
- News modification
- News publication
- News publication cancelation
- News deletion
- Preferences visualization (list)
- Preference visualization (single)
- Preference insertion
- Preference modification
- Preference deletion
- Tag RFID research (Look for tag's details throughout a RFID read)
- Tag SERIAL research (Look for the tag's details throughout its serial number)
- Tag insertion
- Tag modification
- Tag deletion
- Types of payment visualization (list)
- Type of payment visualization (single)
- Type of payment insertion
- Type of payment modification
- Type of payment deletion

- Types of subscriptions visualization (list)
- Type of subscriptions visualization (single)
- Type of subscriptions insertion
- Type of subscriptions modification
- Type of subscriptions deletion

5.2.2 Bar module

The Bar payment module is a part of the IT Beach system by what realizing the RFID tag payment paradigm. The principal objective is the usage of the RFID tag like a debit card, to allow customers to purchase goods and services at the Bar emplacement. This module works together with RIGEL software in order to provide it an RFID tag payment method activating by its functionalities. Our Bar subsystem should In fact interact with RIGEL software in order to complete payments once closed a customer's order. A perfect synchronization should be persecuted. When the Bar employee confirms the order closing and the related payment using RIGEL functionalities, our Bar module would immediately intervene activating the RFID reader to read customer's tag and retrieve all information related to customer's credit account. After the appropriate checks, transaction can be completed.

5.2.3 Turnstiles

The Turnstiles management module is a part of the IT Beach system by what realizing the customers' access procedure. The principal objective is the usage of the RFID tag like an entrance pass, that allows customers to access the structure once they let turnstile reading their RFID tag. Connected to the same local network of one or more turnstiles, its principal task is listening continuously turnstile opening requests and answer them(denying or allowing accesses) after customer's access rights checks.

5.2.4 PDA services module

The PDA services module is intended to be that system provided to monitor employees in order to execute security and customers' checks inside the structure. Equipped with a PDA wireless connected to a DB server, and an integrated RFID reader, he will be free of moving all around the structure being able to sell goods and services, and check instantly all structure activities and a single customer's profile and transactions. We'll now list all the functionalities identified during the Requirements' Analysis phase.

- **login:** this task is needed to get full access to the system. The procedure asks the user to insert the personal credentials (the credentials chosen are as simple as username and password).

- **logout:** : this task is needed to exit safely the system. A logout procedure is needed in order to avoid unauthorized people using PDA services module.

Once the user is logged in, all the functionalities will be available. These ones can be divided in two different areas:

1. **Checking functionalities:** intended to be that functionalities by what the employee is able to check if a customer owns particular access rights or can enjoy a kind of facility, and/or to check positions and usages of valuable assets.
2. **Selling functionalities:** employee can also sell services or subscriptions to particular activities taken in the venue, using the RFID payment system paradigm.

Checking functionalities Once accessed to this area, system offers the following functionalities.

- **PDA client management:** it allows the employee a real time consultation on customers' information. After the customer's RFID tag reading, employee can visualize a customer's profile, his/her access rights and every commercial transaction(services and activity subscriptions purchase) executed during the stay.
- **Services and activities details visualization:** connected to PDA Client Management one, this functionality allows employee to visualize details on services and activities subscriptions eventually purchased by a customer. It's also possible to print paper proves of a customer's purchases.
- **Structure locations monitoring:** real time information updating on customers' transactions allows employee to visualize an updated map of the structure and the locations occupation state with the related details. It will be possible in this way discovering if a customer paid or not for staying in a certain location.
- **Activities monitoring:** it provides a real time monitoring of structure activities involving one or more people, and their rights of participating to them. Activity major details and participants information will be visualized.

Services selling functionalities Once accessed to this area, system offers the following functionality.

- **Services and Activities Selling functionality:** used for a mobile selling of services or activities subscriptions through the RFID tag payment paradigm. Customer's purchases are therefore executed staying comfortably in his/her own location.

Totems

The IT Kiosk is an automatic informative totem placed inside the structure and made up of a touch screen, a RFID reader, a water and dust resistant keyboard, a cash acceptor device and a thermal printer with 80mm wide paper. The IT Kiosk is a device freely usable from the customer, and it's meant to avoid the box office to provide some functionalities in order to make the providing of services quicker. With the IT Kiosk, the customer will be able to:

1. Read news and announcements published by the company.
2. Buy services.
3. Subscribe to entertainment activities.
4. Handle the functionalities related to visualization and modification of data related to previous acquired services and activities
5. Recharge the personal account
6. Handle the personal account thanks to transactions visualizations
7. Handle the personal data (modify personal informations)
8. Modify the personal password related to the wristband.

The customer, in order to access to the functionalities listed above, has to log-in to the system. The log-in come along with the RFID wristband reading. Privacy critical functionalities, such as view of personal data or password modification will require a password in order to be available. We'll give now details about the functionalities listed above.

- **Transactions view and account recharge.** The IT Kiosk provides to the customer the possibility to handle the personal account by accessing to the *Personal Account Management* area. This section gives the possibility to view the personal account status, the transactions made (payments and recharges), the money left and to recharge the account throughout money or coupons. This area is available only after the customer has logged in also with personal password.
- **Visualization of available services and purchase of services.** This section allows the customer to visualize the available services inside the structure and eventually to proceed with the purchasing of one or more of them. The procedure of purchasing a service can only be accomplished by the use of the RFID wristband and not by the use of money. The visualization of the services doesn't require the insertion of the personal password, while the purchasing does.

- **Visualization of available activities and subscription to activities.** This section allows the customer to visualize the available activities inside the structure and eventually subscribe to one or more. The procedure of subscribing to an activity (if it's not a free activity) can only be accomplished by the usage of the RFID wristband and not by the usage of money. The visualization of the available activities doesn't require the log-in with the personal password, while the subscription does.
- **Handle of purchased services and activities.** This section allows the customer to handle the services or activities which have been previously purchased by this. The fruition of a service (e.g. jacuzzi) or an activity (e.g. diving lesson) is controlled by a receipt that has to be printed from the box office or from the IT Kiosk, and this is the purpose of this section. The customer who's bought one or more services or activity can at every time visualize them and decide to print the related receipt.
- **Modification of personal password.** This section allows the customer to modify the password related to the personal account for security issues.
- **Visualization and modification of personal profile.** This section allows the customer to visualize the personal information which were given during the subscription at the box office and eventually to modify them. This sectoin require the log-in with password in order to be accessed.

5.2.5 Non functional requirements

The purpose of this paragraph is to identify and to describe the non-functional requirement of the system. Non-functional requirements are not directly connected with the functional behavior but with how these functionalities are offered to the end user. They refer therefore to quality characteristics. Non funcional requirements include aspects related to:

- Usability
- Maintenance
- Efficiency
- Reliability
- Scalability
- Interoperability
- Policies of product delivery
- Standards and technologies

For each requirement a detailed description and some metrics to verify that the system will respect that constraint will be provided.

Usability

The system will have to interact with different categories of users and under different environment conditions. Each module of the system presents different features so that it's necessary to have a separate analysis for each one. For each module we'll now give an overview on the category of users that will interact with it, in order to give the basis to the usability requirement analysis available on the next paragraph. In the below list we didn't mention the selling point module by purpose, as our system won't introduce any important modification regarding usability. Let's give a brief analysis of the remaining modules:

- **Box office:** this module should be administrated by an employee who is supposed to have a certain knowledge of computers and a certain reliability, as the procedure he's going to be time critical. For this reason it will be a better choice if the employee is a young person, as it's demonstrated that new generation has a better knowledge and confidence with IT systems.
- **Totems:** In this case the end user cannot be related to any category, as we don't know who will use the system. For this reason we considered the typical user for this module as the user who has no background about the use of technology; that's why this is the most critical module from the usability point of view.
- **Remote interface:** this module is supposed to be used by a small number of users encharged to maintain the database updated and by one or two employees who've the right to see the reports regarding crucial data. These user are supposed to have some background with computers as they've been working already in the office so the usability is not a big deal in this module.

Usability requirements As we said above, the modules we're going to develop are enough different to require different usability requirements; anyway, we can find some common requirements to all the modules:

- The time needed to learn how to use the module from the end users shouldn't be more than described in the project plan
- The GUI²of the different modules must follow the guidelines suggested by the usability standards. A description of these guidelines can be required by the sponsor.
- The response times of the system must be highly reasonable (typically inferior to one second) in order to allow a fluid interaction without deadlocks.
- The system should always provide to the user the elements useful to understand what's going on in every instant

²Graphic User Interface

- The software modules have to show a high grade of robustness to the user's input errors (usually with a high rate at least in the first part of the learning period). This implicate the possibility to return to a precedent situation to a certain wrong action or the notification to the user about the fact that an error was made (showing a clear explanation of what was the error about and how to avoid it in future).
- A user manual will be provided. This will document the user about the functionalities offered, about the configuration procedures, about the ordinary management or exceptional one.
- The error messages will have to be clear about the errors in the functionalities of the system. All the errors will be traced in a log file which can be viewed in every moment from the system administrators.

Now let's examine the specific requirements:

- **Box office**

- The clients acceptance procedures must be simple in order to allow a quick queue management; this can be a grave problem in the weekends when the flow of clients requesting services is more than working days. Anyways, the time needed to allow the access into the structure must be the same or less than the one required before the introduction of the system.
- The employee's working station must be equipped with IT instruments compatible with a correct posture.

- **Totems**

- The totems must be placed in sufficiently shadowed places in order to allow a correct visualization of the screen even in the brightest hours of the day as we know that sunlight can disturb the screen visualization in LCD monitors.
- The totems must be equipped with an easily usable keyboard (a possible measurement on the ease of use can be done by an analysis of typing errors).
- Software module for totems must contain an automatic logout in order to avoid malicious access to a profile of an user who's forgot to exit from its private space.

Maintainance and assistance

- The system must be developed to guarantee an easy integration of new software modules. This can be realized throughout an easy installation procedure by the administrator.
- The user manual will have to contain all the operation related to the settings handling by the administrator.

- Maintenance doesn't include modification not related to the functionalities of the system.
- Problems that will cause the non-operativity of the system will have to be solved in the shortest time possible (some hours)

Efficiency

- The system must handle the interoperability of:
 - At least 2 box offices;
 - At least 2 totems;
- Access gates must guarantee an access time not superior to 5 seconds.
- Forms, dialog windows and error messages must be visualized in less than 2 seconds.
- The DBMS³ must be able to handle more than a thousand transaction per second.
- The system will be able to work in PC of average cost for the age 2009.

Portability

- The system is meant to work only in Microsoft Windows XP machines or superior

Reliability

- The system will be able to handle every not fatal exception and to generate immediately an error message to the administrator, recording it into the log file.
- The time to recover from an error has to be less than 30 seconds.
- The percentage of errors must be less than 3 percent.
- The number of defects for line code should be less than 1/1000.

Policies of product deliveries

- The modules of the software must be delivered coherently with what previously indicated in the project schedule in the project plan
- The whole system will have to be installed coherently with what previously indicated in the project schedule; every delay in this phase must be decided in accord with the company's responsible.

³Database Management System

Standard and technologies

- Standards for the RFID devices are the ones indicated by the ISO 14443 and MI-FARE 1k.
- Standard for wireless communication will be WI-FI
- The platform to support the system is Microsoft Windows XP or superior.
- Communication standards for the transport level for the ISO-OSI will be TCP/IP.

5.3 System design

5.3.1 Box office

Design preliminary considerations

First of all, we present some preliminary design considerations that the reader should keep in mind while reading the design choices documented in the next paragraphs.

1. Probably the most important aspect about the procedures that will take place in the box office is that they cannot be considered stable until they get tested against reality. Requests and/or claims by customers or, more likely, managerial directives, could lead to modifications and refining of procedures.
2. Part of the mentioned procedures deal with clients' money. This aspect doesn't have to be underestimated during module design.
3. Even though the box office module will communicate with other modules in an indirect way - making subsystems relatively independent - module's design cannot be performed notwithstanding design choices made for other modules, in order to achieve a high coherence among subsystems.
4. The box office module is intended to be used by one user adequately trained.

Design goals

Subsystem design aims to achieve some fundamental goals, having reference to as many quality attributes belonging to three distinct areas: quality in use, design quality and quality for user. The first area contains those quality attributes that the system has to show during its functioning. Then we have those quality attributes connected to subsystem design and finally the quality attributes bound to user experience while using the system. The quality attributes that the system has to expose while in use derives directly from the non-functional requirements documented in the RAD of this project and summarized in this chapter at paragraph 5.2.5. Let's see more in details what design goals are module specific instead.

Quality in use. Considering the high impact that could have an event putting this module unavailable (even higher in case of long lasting unavailability) - let's think to the queue of customers waiting for check in - one of the fundamental goals is module **Availability**. This is intended as the amount of time during which the module is operative and can be measured, inversely, as the percentage of time during which the system is unavailable on a preset period. Keeping in mind this goal will minimize as much as possible the events capable to put the module unavailable.

From the goal just mentioned, another one derives quite naturally: module **Reliability**. The box office module will have to be available as much as possible, but has also to complete requested tasks correctly for all the time in which it results available.

Reliability is tightly coupled to the level of module **Security** achievable, since some malicious users could try to take advantage from a malfunctioning. The presence of procedures dealing with clients' money increases drastically the security level required to this module, so that module design cannot avoid to make deep considerations about possible vulnerable points that has to be secured.

Design quality. The module has to take into consideration the following quality attributes, putting their achievement among its design goals:

- **Flexibility:** it is related to the ability of the module to adapt to changes imposed to internal policies and procedures. The module has to be easily reconfigured to support eventual (and probable) modifications to the procedures now imagined for the box office.
- **Maintainability:** this term refers to the ability of the module to undergo some changes to its components, services, features and interfaces. These modifications are tightly coupled to the adding and modification of module's functionalities, bug fixing and changes explained in the previous point. Along with flexibility, maintainability is a fundamental design goal.

Quality for users. Precise considerations regarding the type of module-user interaction imagined brought to the awareness of the importance of a pleasant and fluid interaction, most of all because of procedures' length and complexity (i.e. the check in of a big group of persons). All the solutions derived from this awareness combine to bring to a high level of module **Usability**, considered as an essential goal for every software.

Application structure

We present here an overview of how the box office module have to be structured in terms of type of application, deployment of fundamentals parts, architectural styles and selected devices.

Type of application The application thought to support box office's procedures will follow a specific type known in literature as *Rich Client Application*. The reasons why

we've chosen this type of application derive naturally from the type of procedures to support, from the development environment selected and from the devices with which the user will have to interact (a desktop pc). In fig. 5.3, we propose a schema that should better explain the features of the chosen type of application.

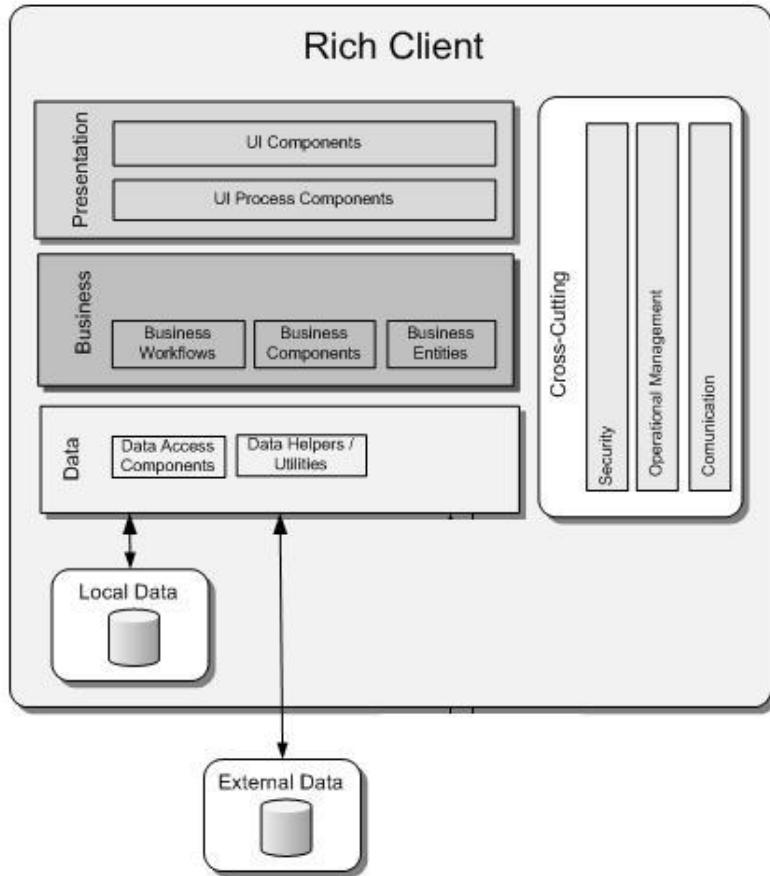


Figure 5.3. Structure of Rich Client

Deployment constraints The deployment hypothesized for this module is known as non-distributed, meaning that all the components necessary for user interaction and for business logic will reside on a single machine.

Architectural style The architectural style that resulted to be more suitable to the functionalities intended for the box office is known in literature as *Layered*. This model should consent to face subsystem's complexity, grouping the several functionalities in a certain number of competence areas. This style should also increase the level of maintainability and extensibility, minimizing dependencies. In particular, the most appropriate

schema results to be the widely spread three-layered partitioning: Presentation Layer, Business Layer and Data Access Layer.

Technology selection Considering the assumptions made regarding the developing environment (.NET and Visual Studio 2008), the development technologies employable are limited: Windows Forms and WPF (Windows Presentation Foundation). The second has a very high innovative component and, since the graphical requirements of this module are not excessive, it's not opportune to face the issues related to WPF applications development. Our choice then had to fall on **Windows Forms** technology, able to support design and development of applications offering good performances and interactivity.

Design key hotspots Design phase takes into account a certain number of key features particularly relevant for the achievement of targeted goals. These key hotspots are connected to some quality attributes and a series of choices concerning cross-layering issues (that is regarding all architectural layers). Starting from relevant quality attributes for the box office module these key features have been identified:

- Availability;
- Flexibility;
- Manageability;
- Scalability;
- Security;
- Usability;

Cross-layering issues indicate the presence of the following design key features:

- Authentication and authorization;
- Caching and state
- Communication
- Exception management
- User experience
- Validation
- Workflow

For each one of the key features listed some considerations will be made with regard to the solutions to adopt towards an effective design.

Availability. Module's availability is threatened by a series of events:

- Attacks from outside the system (i.e. a Denial Of Service attack);
- Database unresponsiveness;
- Network unavailability;
- Long stall on a crucial resource;
- Bugs and errors in the module.

The minimization of attacks' probability concerns security requirements that will be explored in the relative paragraph. Since database unavailability would have a dramatic impact on module functioning, the necessary information will have to reside on a database accessible on a well designed local network. IT Be@ch has only one shared resource, the database. That's why long stall issues are limited to locks posed on one or more tables (or records) in the database. Query to database will have to be optimized avoiding long lasting locks on data.

Flexibility. Module design will have to strictly follow layers decomposition. This means that particular attention has to be dedicated when categorizing a component as belonging to a layer rather than another, in order to achieve an adequate level of coupling and cohesion. This will make easier to substitute or expand user interface, or make some modifications to business procedures or data access schemas.

Manageability. Module's "health" should be monitored through a log mechanism coupled with notes to system administrator. Logging procedures will have to track all relevant episodes (especially if error conditions).

Scalability. Though until now each beach resort has only one box office, it is quite easy to imagine a context with more than one computer selling subscriptions and daily accesses. The presence of more than one box office working simultaneously imposes particular attention to database access, as this is the only synchronization point.

Security. Module's security is strictly connected to these two features:

- Vulnerable module's points: malicious attacks could take place through most vulnerable points of the system. That's why the design should reduce as much as possible the number of those points and beware of the geographical network that the system should be connected to.
- Personal data protection: personal data should be stored only in one or more DB servers and not replicated throughout other nodes of the network. The accessibility of those data has to be restricted to trusted users in order to respect clients' privacy.

Usability. Module's main usability issues concern procedures' complexity and length (i.e. new subscription and daily access procedures) and the simultaneous presence of many information in a little number of views.

Authentication and authorization. It will be necessary to identify those areas for which the user will have to be authenticated and authorized in order to access. Authentication will be based on the input of user credentials like username and password. Work session will remain valid until a formal log out is performed by the user or until module shut down. The authentication typically required by Microsoft Windows Operative Systems will not be enough; an authentication specific to IT Be@ch will be needed.

Caching and state. Box office's module will have to manipulate quite static data towards the working day (i.e. available subscriptions type or categories of client). That's why it would be preferable to keep those data in a local cache to be synchronized with database only in specific occasions. Module's performance evaluation will tell if a more sophisticated caching mechanism could be useful, like through the observation of most used data or data access patterns. The subsystem is not intended to work without database connectivity. That's why is not necessary to consider working scenarios when it's impossible to update the database promptly. Transient data, that is those data that are protean while client is working, will have to be synchronized and only after the synchronization data can be considered reliable. The concurrency model has to be pessimistic consisting in data locking by the client that intends to access them. The adoption of this concurrency model imposes the optimization of query execution time with the target of minimizing the amount of time for which a resource results busy for other requests. State will include the information shown in the user interface and data necessary in multi-step procedures (i.e. new subscription procedure). For these data a memorization mechanism has to be arranged.

Communication. Box office module will have to communicate only with database. Database connection don't require particular considerations since selected technologies present a high degree of maturity. It's anyway important to put some security logic on personal data passage through the network to avoid sniffing occurrence.

Exceptions management. During its functioning the system could throw a certain number of exceptions that our team should prevent or eventually manage. Some exceptions could lead to the immediate system block; others could simply introduce one or more errors dangerous for the successive correct functioning. In any case it's opportune that all erroneous conditions comes on a log file. Errors notifications don't have to report any restricted information, but only a description of the occurred error and eventually a brief recovery procedure.

User experience. This term refers to a broad range of aspects directly connected to module's usability and more precisely to user satisfaction during the interaction with

the module. This is in fact the third component of usability after efficacy and efficiency. The high importance of this aspect derives from the long lasting interaction required (several hours uninterrupted). First considerations concern working environment ergonomics. Then user interface ergonomics will have to be analyzed. The amount of effort spent during requirements analysis phase to optimize operative procedures leaves few chance to a more easy approach without renouncing to a certain degree of flexibility. The achievement of a pleasant interaction is thus dependant to the pursue of a series of usability guidelines quite consolidated in the field of Rich Client Application development through Windows Forms. The primary goal is to make the interaction the less possible tiring and tedious, putting aside some others usability factors less significant for this module.

Validation. User input validation has to be attentively performed in order to avoid malicious attacks to the system (i.e. SQL injections). All text fields will have to be validated before passing the input to the business layer. The validation will be performed through the use of regular expressions and the field intended for generic text will have to be sanitized by eliminating potentially dangerous characters (

Workflow. New membership subscription and daily access follow a procedure divided into multiple steps with a Wizard style that helps user in defining the several features of the subscription. The chosen pattern is known as Use Case Controller that comply perfectly with the roles division shown during requirements elicitation and analysis. A controller object is in charge of keeping the right sequence of steps and maintaining temporary state of user choices throughout steps making it possible to navigate through the wizard and modify inserted data.

Chosen architectural style

According to the considerations made until now, we propose in fig. 5.4 the architecture that our team will have to keep in mind while developing the box office module.

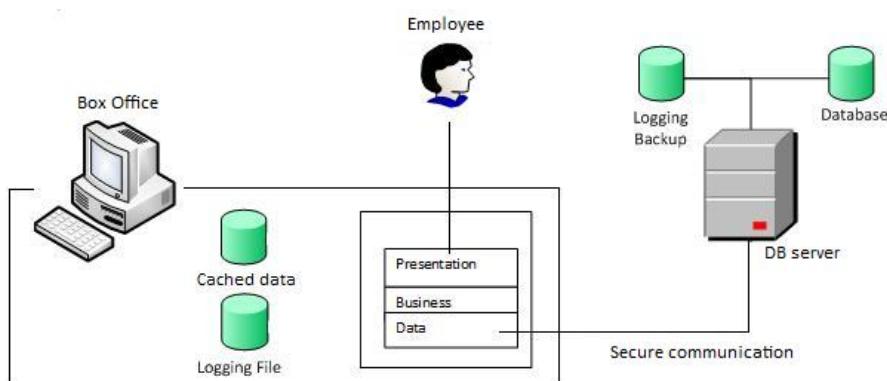


Figure 5.4. Box Office's candidate solution

5.3.2 Remote interface

Preliminary design considerations

We'll start the design description of the Remote Interface module with some considerations about the application type of the module itself, in order to define some characteristics about the subsystem design:

- The module will be installed in a personal computer located in the central office of the MIIB and will be used only by a restricted number of trained employees.
- The module doesn't need a constant connection with the remote DB server, as it works with a local database which has to be synchronized with the remote one.
- The informations obtainable through the module are to be considered reserved; only authorized employees will be able to access them.
- The module won't be able to provide real time information about what's going on in the Beach, but will handle information of the day before as the synchronization of the local database with the remote one happens once in a day.

Design goals

Let's see in details which design goals are to be considered fundamental for the Remote Interface module.

Quality in use Reliability: the module must be highly reliable especially because of the impact that an error which leaves the database in a not consistent state could have when the database are synchronized (imagine the price of a subscription set to zero for an exception not handled). Nevertheless it's important to stress that this module is more tollerant to the time needed to repair the errore, as it's not directly connected with the activity in the beach: while an error in the Box Office module has to be solved in the shortest time possible (possibly minutes or an hour), as the synchronization of the local database with the remote one will happen in night time, an error in the Remote Interface can be not so time critical. The repair must happen before the next synchronization with the remote database.

Design quality

- Flexibility: it consists in the capacity of the module to be adapted to the changes imposed by the company's policies to the procedures. The subsystem must be therefore easily reconfigurable in order to support in a complete way the eventual (and actually very probable) changes to the informations handled and to the procedures prodived by the module we're examining.

- Manutenability: it consists in the capacity of the module to be modified in its functionalities, in its components, in its services and its interfaces without affecting the whole functionality of the system. These modifications are directly connected with the adding or the modification of one or more functionalities, with the correction of bugs and defects or to the changes we described in the previous point. Manutenability is a key design goal for the remote interface.

Quality for the user

- Usability: the Remote Interface, as we described in the previous paragraph, will handle a large number of informations as it will be the interface used to fill the database. Usability will be made hard to handle from the number of informations that will be showed to the user: for this reason, it will be developed an intuitive hierarchy in order to allow the employee to easily find the desired functionality.

Application Structure

We present here an overview of how the Reemote Interface module that have to be structured in terms of type of application, deployment of fundamentals parts, architectural styles and selected devices.

Type of application The type of application chosen for the Remote Interface is **Smart Client Application**. Since this is the same application chosen for the Box Office module, we will refer to its description in the paragraph related to the chosen application of the Box Office.

Deployment constraints The deployment hypothesized for this module is known as non-distributed, meaning that all the components necessary for user interaction and for business logic will reside on a single machine.

Architectural style The architectural style that resulted to be more suitable to the functionalities intended for the box office is known in literature as Layered. This model should consent to face subsystem's complexity, grouping the several functionalities in a certain number of competence areas. This style should also increase the level of maintainability and extensibility, minimizing dependencies. In particular, the most appropriate schema results to be the widely spread three-layered partitioning: Presentation Layer, Business Layer and Data Access Layer.

Technology selection Considering the assumptions made regarding the developing environment (.NET and Visual Studio 2008), the development technologies employable are limited: Windows Forms and WPF (Windows Presentation Foundation). The second has a very high innovative component and, since the graphical requirements of this module

are not excessive, it's not opportune to face the issues related to WPF applications development. Our choice then had to fall on Windows Forms technology, able to support design and development of applications offering good performances and interactivity.

Design key hotspots

The design phase takes into consideration a list of key points or particular relevance in order to reach goals described before. The key point are connected with some quality attributes and with some features about transversal issues (that are linked with all the layers of the architecture). Starting from the quality attributes, we identified the following key points:

- Flexibility
- Security
- Usability

The transversal issues indicate the presence of the following key points of design:

- Authentication and authorization
- Caching and state
- Communication
- Exception handling
- Logging
- User experience
- Validation

Most of the key points above listed are in common with the Box Office module; for this reason we'll avoid describing them again. Nevertheless, some key points have got particular issues that we're going to see below:

- **Flexibility:** the flexibility required from the module is high as the Remote Interface should be able to handle mutable information and procedures. This is why it's important to adopt a development which allows the substitution and the extension of the functionalities what made up the module. The layering approach tends to promote the flexibility of the considered subsystem.
- **Security:** the Remote Interface module must adopt a particular security policy especially in the protection of the reports, as they contains reserved and sensitive information.

- **Usability:** the most important aspect regarding usability in the Remote Interface regards the number of functionalities accessible from a single view. Too many buttons in the same window would make the module hard to deal with: that's why, as we already said above, the module must be organized in a hierarchical way in order to differentiate the functionalities and therefore to present only a limited number of buttons in each window.
- **Authentication:** authentication to the system will be based on the immission of personal credential such as username and password. The session will be valid until explicit logout made by the employee or till the exit of the system.
- **Caching:** the Remote Interface, since it is going to be developed as a Smart Client Application, will be able to work offline. The scenario is made simpler by the presence of only one client application handling the data (for this reason there won't be any concurrency policies applied). Since the database is local and the modifications applied to it would be visible the following day after they're made, the application can work also in case of no connectivity. The synchronization between the local and the remote database will have to happen in the time span when the beach resort is not opened to the customers (realistically during the night hours). In order to simplify the reporting operations, the available data will be considered as the ones referring to the previous day.
- **Communication:** the communication between the Remote Interface and the server containing the remote database can be only in one direction: from the Remote Interface client to the remote server. Therefore, the connection doesn't require any particular development thanks to the maturity of the technologies selected. It will be important, anyway, to use security technologies such as SSL⁴, in order to avoid network attacks (like sniffing or IP spoofing).

Chosen architectural style

The chosen architectural style for the Remote Interface is the layered model with three layers: data layer, business layer and presentation layer. The chosen layer is the same as the Box Office one. In fig. 5.5 we show an image representing the final architecture for the Remote Interface module.

5.3.3 Bar payment module

The Bar payment module is a part of the system implementing the RFID tag payment paradigm. The main aim is the usage of the RFID tag like a debit card, to allow customers to purchase goods and services at the Bar emplacement. This module can be deployed and replicated at one or more Bar selling-points and implies, for each emplacement, the usage

⁴Security Sockets Layer: it's a cryptographic protocol which allows a secure communication and an integrity check in the data transmitted on TCP/IP networks such as the Internet. SSL encrypts the communication from the source to the destination (end-to-end) on the transport layer.

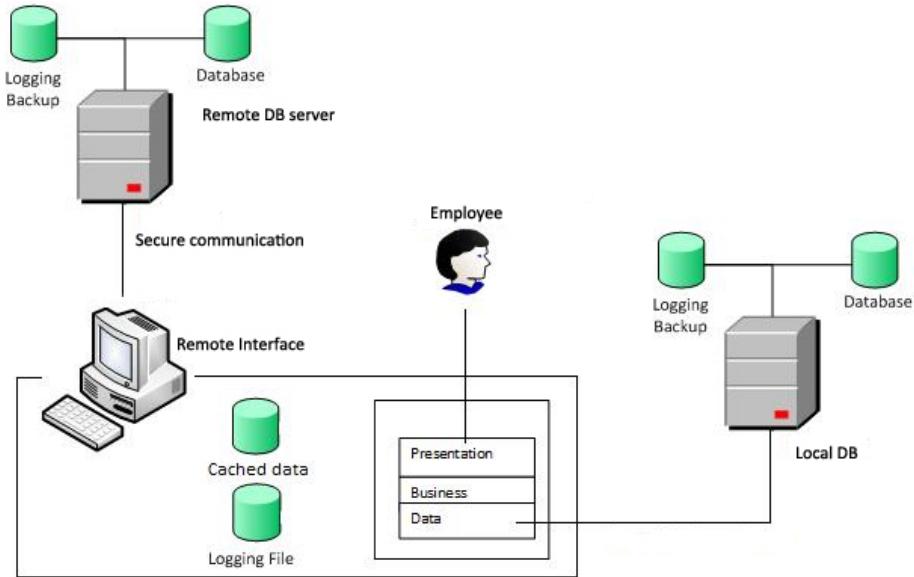


Figure 5.5. Remote interface architecture

of a PC, an RFID reader and a connection to the DB server. The payment procedure is the same as a credit or debit card one is:

- A Customer asks to pay goods purchased, and after a PIN insertion and tag reading confirms the payment
- The system will execute the appropriate checks decrementing, if every check was positively completed, his/her virtual account

Design preliminary considerations

A careful SW and HW Requirements' Analysis brought our study group to define some preliminary design considerations that the reader should keep in mind while reading the design choices documented in the next paragraphs.

- The subsystem's should collaborate with another Bar management software chosen to manage Bar orders and fiscal registration. RIGEL software, realized by APS s.n.c., will be the one in charge of this task. This software allows the Bar selling-points to manage goods orders , tables and register all the payments. Additional RIGEL PDA modules enable to make mobile orders quickly and without any usage of paper. The only method of payment RIGEL offers is the cash or coins one. Since our objective is to enlarge RIGEL functionality with a new payment method based on the customer's RFID tag, used like a debit card, a strict collaboration has to be persecuted, in order to interface and synchronizing appropriately our Bar and RIGEL modules.

- The sub system's core will work on a stand-alone emplacement together with RIGEL, while the DB, from what it will take the customers' credit account information, will be installed on a different machine and shared also by the other subsystems. Being independent from other subsystems, designer should not take care of other subsystems' features, although he/she should follow some design policies established(names, development styles, framework etc.).

Design goals

Subsystem design aims to achieve some fundamental goals, having reference to as many quality attributes belonging to three distinct areas: quality in use, design quality and quality for user.

Quality in use Considering the high impact that could have an event putting this module unavailable - let's think to the queue of customers waiting for paying and the related inconveniences in case they don't own cash or coins- one of the fundamental goals is module **Availability**. It is intended to be the amount of time during what the module is operative and can be measured, inversely, as the percentage of time during which the system is unavailable on a preset period. Keeping in mind this goal will minimize as much as possible the events capable to put the module unavailable.

But **Availability**, does not imply time working continuity. A system could result operative for the 90 percent of the time in what it is under observation, but being really working only in little time intervals interrupted form continuous and instant malfunctions. Another important design goal is therefore the module **Reliability**. It can be intended to be the system ability in being continuously operative, in order to execute correctly every tasks that have to be done during all that time in what it results available. Guaranteeing Reliability means trying to minimize the malfunctions/faults amount that could bring the system to an inconsistent status or crash condition.

Another important quality attribute, considering the previous preliminary design considerations, will be the module **Interoperability**.It is intended to be the ability of diverse components of a system or different systems to operate successfully by exchanging information, often by using services. An interoperable system makes it easier to exchange and reuse information internally as well as externally. Our Bar subsystem should In fact interact with RIGEL software in order to complete payments once closed a customer's order. A perfect synchronization should be persecuted. When the Bar employee confirms the order closing and the related payment using RIGEL functionalities, our Bar module would immediately intervene activating the RFID reader to read customer's tag and retrieve all information related to customer's credit account. After the appropriate checks, transaction can be completed.

Communicating to DBMS to manage sensitive operations like payments and transmitting customers' credentials, Bar payment module should be capable of deploying efficient **Security** countermeasures to avoid that any attacker could cause faults and/or malfunctions. Therefore, particular attention is needed on boundary components' design, sensitive access points and components representing specific bottlenecks.

Module **Performance** is the last quality attribute to be achieved in order to avoid long queues at bar selling-points. The subsystem should be able to retrieve customers' credit account information in a few seconds after reading their RFID tag. Considering the simple task to be accomplished(it requires just a query passing a tag UID) it seems there couldn't be performance troubles.

Design quality The design quality is the same as the Box Office module.

Application structure

Identified design goals to achieve, let's continue on defining a design overview of Bar subsystem considering the following analysis steps:

- Determination of the application typology according to requirements and application constraints
- Identification of deployment constraints
- Choice of the architectural style
- Determination of technologies to deploy

Type of application The Bar module realization implies a strict interaction with the RIGEL software in order to provide it the RFID tag payment functionality. Considering it doesn't require any user interface layer, complex algorithms and any user interactions, offering only one simple functionality it can be realized like a Service application. Infact, a service application is a public interface that provides access to a unit of functionality, it literally provides one programmatic "service" to the caller, who consumes them. It offers a slim and simple architecture that guarantees performance and high availability. Without any user interaction needed, it can start automatically at the OS bootstrap(if automatic start modality is activated) to be immediately used by any other systems like the RIGEL one. In figure 5.6 we show the major features of a service application.

Deployment constraints As already mentioned, Bar payment module is a component of the IT Beach system whose architecture is a centralized repository type. Infact, every subsystem should communicate with a remote DB to gather the information necessary to execute their functionalities. Considering this deployment constraint, it's necessary keeping in mind that the Bar subsystem should be provided with a remote communication paradigm for a correct and complete running. Every Bar module should be deployed on a single machine together with RIGEL software, and installed like a service starting automatically at the OS bootstrap. DB remote communication will use a network interface to connect to a local network. RIGEL -Bar module communication should be based on messages since they communicate on the local machine. Therefore, transport and message protection decisions must be made based upon local machine trust boundaries and users. The deployment architecture chosen is the not-distributed type and, whose functioning scheme is shown in the fig. 5.7.

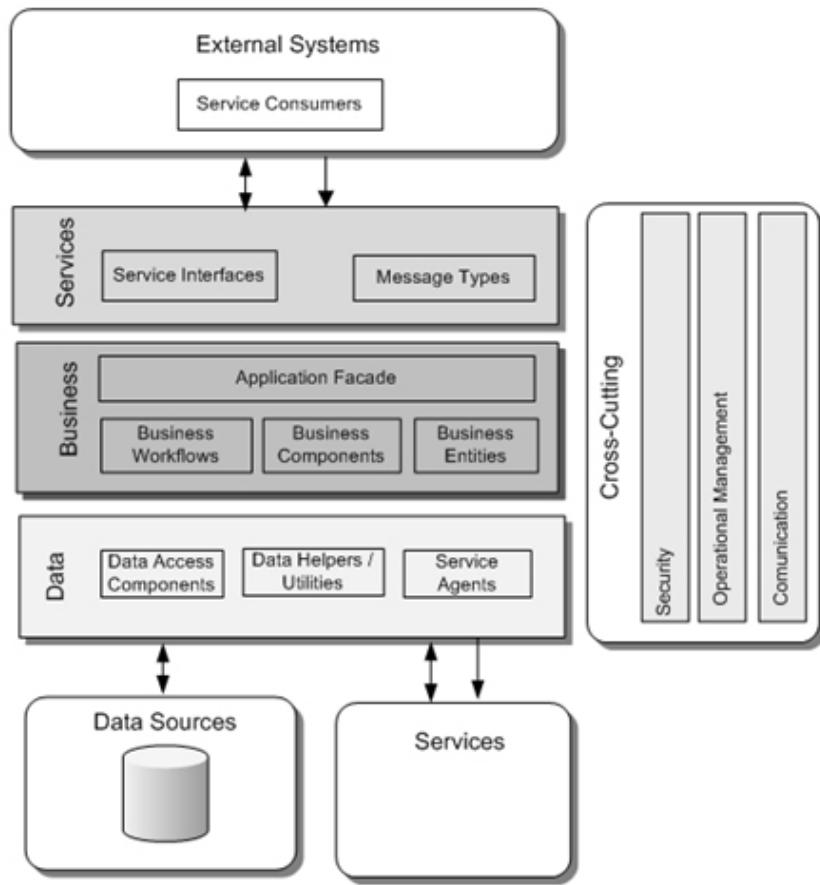


Figure 5.6. Structure of typical service application

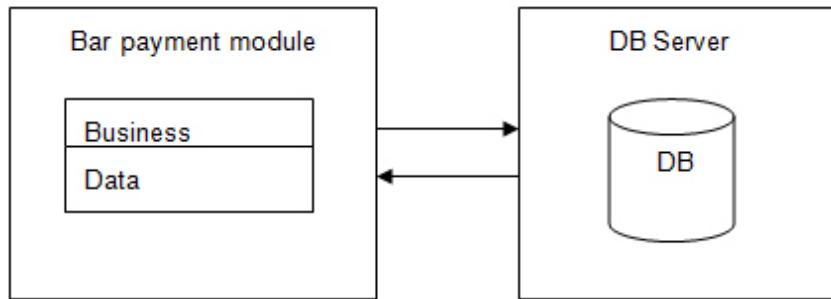


Figure 5.7. Deployment features for the BAR payment module

Architectural style Basing on the experience of the design and implementation team, architectural style chosen resulted to be the one named in literature ”“2-Layered”. This architectural style is characterized by a 2 logic levels decomposition that are:

- Business logic, logic level that components dedicated to manage application logic core in terms of data processing, algorithms execution.
- Data Access, logic level grouping all components related to manage data entities and local or remote DB access procedures.

No Presentation layer is considered since Bar payment subsystem will be implemented like a Service Application that offers the RFID tag payment functionality that will be used by another software system. Therefore, it does not require any user interaction. Layered structure guarantees high Maintainability and extensibility thanks to the decomposition of system elements in different areas of competence. Subsystem results, therefore, easily modifiable and manageable in case of future necessary modifying. Layered decomposition allows to realize slim, simple system architecture, simply manageable in short and long terms, satisfying in this way the many design goals treated in the previous paragraphs.

Technology considerations Strictly connected to the application typology is the technology to deploy. Windows Service technology resulted to be the most appropriate and advanced regarding service applications development in Microsoft OS. We'll therefore utilize .Net framework and C# language to develop Bar payment subsystem components.

Key Hot spots

Design phase considers many relevant and necessary key points to achieve the previous set design goals. They concern to many quality attributes and a certain number of "transverse choices" (that involve every layer of the architecture). Starting from the quality attributes key points we find:

- Performance
- Security
- Scalability
- Availability

Transverse choices imply the presence of the following design key points:

- Communication, considering sensitive data transmitted, related to payments or customers' credentials, every end-to-end communication (RIGEL-Bar module and Bar Module-DB server) should be encrypted and authenticated as principal countermeasures to sniffing and spoofing attacks.
- Concurrency and transactions, read and write locking DB mechanisms should be implemented to satisfy ACID properties of DB transactions.
- Exceptions management and logging, a precise mechanism to raise and log exceptions from lower to higher layers has to be implemented in order to reduce the probability that the system could crash without any notify.

Candidate solution

The design study and choices completed in the previous paragraphs can be reassumed in the figure 5.8.

Description: Bar payment module is characterized by a layered structure and runs on a machine together with RIGEL software as an automatic Service Application. It continuously listens to a RFID tag payment procedure to be started by the employee using RIGEL order closing functionalities. It reads the RFID tag and makes the appropriate checks to execute a purchase transaction. Every error or system faults raising an exception is instantly logged in one or more text files.

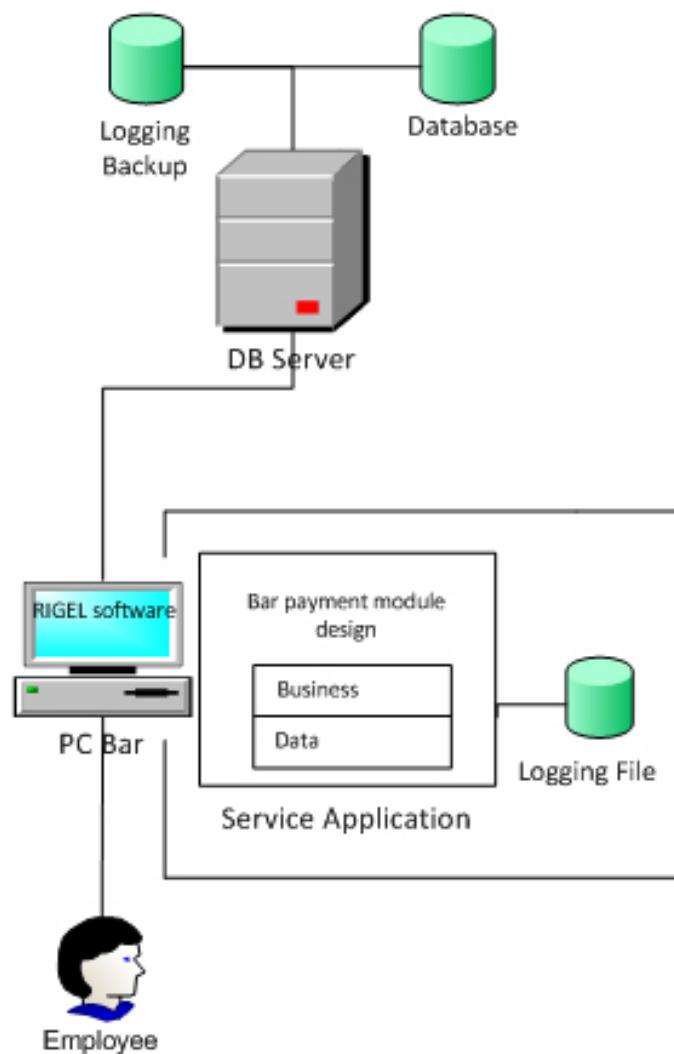


Figure 5.8. Candidate solution for BAR module

5.3.4 Turnstiles management module

The Turnstiles management module is a part of the IT Beach system by what realizing the customers' access procedure. The principal objective is the usage of the RFID tag like an entrance pass, that allows customers to access the structure once they let turnstile reading their RFID tag. Its realization requires one or more turnstiles with an integrated RFID reader, each connected to a local network. Every time a customer passes his/her RFID tag on the related reader, the turnstile onboard chip communicates with the turnstile management module located on the same local network transmitting the UID read. After the appropriate checks on the DB server, it denies or allows customer's access.

Design preliminary considerations

A careful SW and HW Requirements' Analysis brought our study group to define some preliminary design considerations that the reader should keep in mind while reading the design choices documented in the next paragraphs.

- Since no user interaction and other subsystem communication is needed, the subsystem's core can be installed on the same machine containing customers subscription data, the DB Server. Therefore, communication to turnstiles, everyone containing a specific Ethernet network interface, will be performed through their connection to the local network. Being independent from other subsystems, designer should not take care of other subsystems' features, although he should follow some design policies established(names, development styles, framework etc.).
- Turnstiles management module is a singleton subsystem, it means that only one instance of this module can be deployed for the whole IT Beach system in order to avoid redundant operations. Turnstiles messages cannot be duplicated to be sent to many turnstile management modules, otherwise an eventual turnstile opening could be performed many times for a single customer's access. Only one turnstile module will process every turnstile message.

Design goals

Subsystem design aims to achieve some fundamental goals, having reference to as many quality attributes belonging to three distinct areas: quality in use, design quality and quality for user.

Quality in use Customers' access procedure is the principal task to be performed by IT Beach system. After a subscription purchase customer should be allowed to enter the venue easily. Eventual malfunctions, especially in medium-big dimension structures, could cause queues and wastes of time annoying customers' stay at start. One of the fundamental goals is thus module **Availability**. It is intended to be the amount of time during which the module is operative and can be measured, inversely, as the percentage of time during which the system is unavailable on a preset period. Keeping in mind this goal will minimize as much as possible the events capable to put the module unavailable.

Tightly coupled to the previous one is module **Reliability**. Intended to be the ability of a system to remain operational over time, it requires that turnstile management module should be continuously available without any instant error or faults. Although it seems not critical, instant error occurred during the customer's tag reading, lets the system appearing not working, causing every inconvenience related to this status.

Module **Performance** is the last quality attribute to be achieved in order to avoid long queues at turnstiles. The subsystem should be able to gather messages(containing tag UID) sent by every Turnstile connected to the network, retrieve customers' subscription information, and activate or not turnstile opening mechanism . Considering the simple task to be accomplished(it requires just a query passing a tag UID) it seems there couldn't be software performance troubles.

IT Be@ch could deploy one or more turnstiles, therefore it's necessary to consider every problem connected to the adding of one or more turnstiles. Designer has to necessarily guarantee high **Scalability**, intended to be the system capability to adapt(without introducing any malfunction) to every unpremeditated architectural(HW or SW) change(expansions/contractions in short and long terms.

Application structure

Identified design goals to achieve, let's continue on defining a design overview of Bar subsystem considering the following analysis steps:

- Determination of the application typology according to requirements and application constraints
- Identification of deployment constraints
- Choice of the architectural style
- Determination of technologies to deploy

Type of application Turnstile management module is ideated to execute only a specific task characterized of few activities:

- Listening turnstiles requests/messages containing RFID tag UID read once customer tries to access the venue.
- Retrieve customer's subscription information through that customer's tag UID
- Answering appropriately(denying or allowing access to client) to every turnstile related to an opening request

Considering these simple task, that no user interaction is needed and the necessity of guaranteeing continuous operability once the system is activated, we could develop turnstile management module as a Service Application. Infact, a service application is a public interface that provides access to a unit of functionality offering a slim and simple architecture that guarantees performance and high availability. Without any user interaction

needed(it does not offer any graphical instrument), it can start automatically at the OS bootstrap(if automatic starting modality is activated) to be immediately ready. These features allow us to perfectly satisfy every design goal identified and easily accomplish our task.

Deployment constraints As already mentioned in preliminary design considerations, only one turnstile management module should be deployed for the IT Beach customer's access procedure. Since no user interaction and other subsystem communication is needed, it can be deployed directly on the DB server and communicating through the local network to the access devices(turnstiles).

It has to be installed like a service starting automatically at the OS bootstrap. Turnstiles-Turnstile management module communication, should be based on UDP-TCP messages since they run on different machines. Therefore, transport and message protection decisions must be made based upon TCP or UDP protocols.

The deployment architecture chosen is the not-distributed type and, whose functioning scheme is shown in figure 5.9.

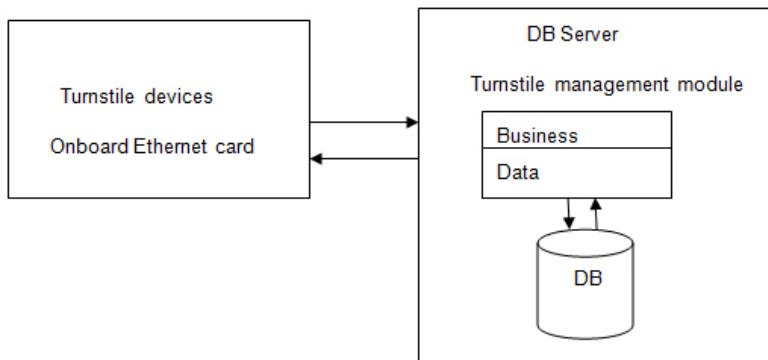


Figure 5.9. Deployment architecture for the turnstile module

Architectural style Basing on the experience of the design and implementation team, architectural style chosen resulted to be the one named in literature "2-Layered". This architectural style is characterized by a 2 logic levels decomposition that are:

- Business logic, logic level that components dedicated to manage application logic core in terms of data processing, algorithms execution.
- Data Access, logic level grouping all components related to manage data entities and local or remote DB access procedures. No Presentation layer is considered since turnstile management subsystem will be implemented like a Service Application that manages the customers access procedure automatically without interacting to any user.

Layered structure guarantees high Maintainability and extensibility thanks to the decomposition of system elements in different areas of competence. Subsystem results, therefore, easily modifiable and manageable in case of future necessary modifying. Layered decomposition allows to realize slim, simple system architecture, simply manageable in short and long terms, satisfying in this way the many design goals treated in the previous paragraphs.

Technology selection Strictly connected to the application typology is the technology to deploy. Windows Service technology resulted to be the most appropriate and advanced regarding service applications development in Microsoft OS. We'll therefore utilize .Net framework and C# language to develop turnstile management module subsystem components.

Key Hot spots

Design phase considers many relevant and necessary key points to achieve the previous set design goals. They concern to many quality attributes and a certain number of "transverse choices"(that involve every layer of the architecture). Starting from the quality attributes key points we find:

- Performance
- Scalability
- Availability

Transverse choices imply the presence of the following design key points:

- Concurrency and transactions, read and write locking DB mechanisms should be implemented to satisfy ACID properties of DB transactions.
- Exceptions management and logging, a precise mechanism to raise and log exceptions from lower to higher layers has to be implemented in order to reduce the probability that the system could crash without any notify.

Candidate solution

The design study and choices completed in the previous paragraphs can be reassumed in the figure 5.10.

Description: Turnstile management module is characterized by a layered structure and runs on the same machine containing the database, the DB Server. It continuously listens turnstile opening requests and answer them(denying allowing accesses) after customer's access rights checks. Every error or system faults raising an exception is instantly logged in one or more text files.

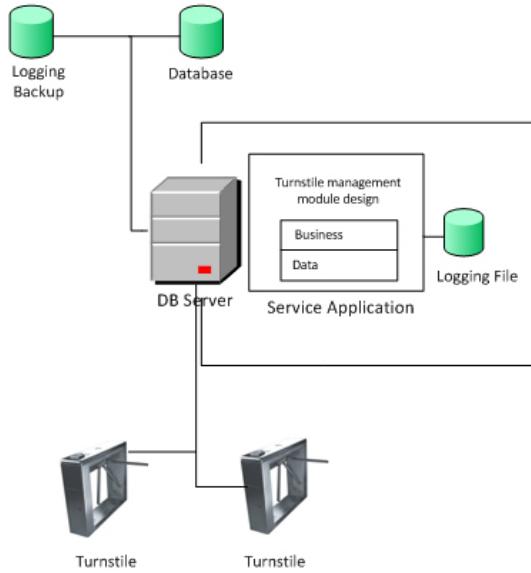


Figure 5.10. Candidate solution for the turnstile module

5.3.5 IT Kiosk

The IT Kiosk subsystem, part of ITBe@ch IS, is a solution destined to all structure's customers. It is intended to be a computerized emplacement characterized of a touch screen monitor, an RFID reader, a keypad (water and weather proof), a money inserter device and a 60mm thermal printer. IT Kiosk is defined on customer's needs and allows him/her to:

1. Visualize corporate news and communications
2. Purchase services
3. Subscribe to entertainment or sport activities
4. Manage Services and Activities Portfolio
5. Manage(visualizing and/or recharging by money) a personal credit account usable by an RFID tag
6. Modify his/her own password for services and goods purchasing

Before passing to the IT Kiosk sub-system design it's necessary to define utilization, technological and installation assumptions, constraints and quality attributes.

Design goals

Subsystem design aims to achieve some fundamental goals, having reference to as many quality attributes belonging to three distinct areas: quality in use, design quality and quality for user.

Quality in use Considering the important roles covered by IT Kiosk together with the PDA module in selling services and activities subscriptions, and together with Box Office for the customer's credit account recharging, eventual IT Kiosk malfunctions/unavailability could lead many inconveniences to the fluid system running. Therefore, one of the most important goals is the module **Availability**, intended to be the amount of time during which the module is operative and can be measured, inversely, as the percentage of time during which the system is unavailable on a preset period. Keeping in mind this goal will minimize as much as possible the events capable to put the module unavailable. IT Kiosk malfunctions would imply the increase of other two subsystems' workload, In fact:

- The Box Office one, is defined not only to execute principally customers' subscriptions registration but also to recharge their virtual credit account only during the first access.
- The PDA module, is used not only for executing a real time monitoring f the structure but also to sell services and activities subscriptions.

Keeping in mind this design goal will allow us to perfectly face and minimize that events that responsible of its unavailability.

But Availability, does not imply time working continuity. A system could result operative for the 90 percent of the time in what it is under observation, but being really working only in little time intervals interrupted form continuous and instant malfunctions. Another important design goal is therefore the module **Reliability**. It can be intended to be the system ability in being continuously operative, in order to execute correctly every tasks that have to be done during all that time in what it results available. Guaranteeing Reliability means trying to minimize the malfunctions/faults amount that could bring the system to an inconsistent status or crash condition.

Communicating to DBMS and managing sensitive operations like payments, recharges, and transmitting customers' credentials, IT Kiosk should be capable of deploying efficient **Security** countermeasures to avoid that any attacker could cause faults and/or malfunctions. Therefore, particular attention is needed on boundary components' design, sensitive access points and components representing specific bottlenecks.

As already mentioned in the paragraph Design preliminary considerations, users expected to interact with the IT Kiosk are characterized from low-medium technological skills and a low predisposition to IT usage. For this reason , every fault or malfunction could result annoying, increasing also customer's resistence to change. Information visualization delays result also more annoying to users, In fact we can just think to our reaction every time the internet connection is slow in visualizing web pages. Although goals like Availability and Reliability try to minimize possible malfunctions, certainly they don't focus on application answer delays. Therefore, another important goal to achieve is the **Performance**. IT Kiosk should answer in the lowest time to allow a fluid visualization and to involve as much as possible people in using it. A simple and slim design could therefore help in reaching high performances. In some cases a decrease of performances is related to a busy network, but assuming to realize a local network whose traffic is characterized principally by few DB queries there should not be problems from this point of

view.

IT Be@ch could deploy one or more IT Kiosk subsystems, therefore it's necessary to consider every problem connected to the presence of duplicated IT Kiosk modules. Designer has to necessarily guarantee high **Scalability**, intended to be the system capability to adapt (without introducing any malfunction) to every unpremeditated architectural (HW or SW) change (expansions/contractions) in short and long terms. Using more than one IT Kiosk module means meditating a careful data access control mechanism to the shared DB server and guaranteeing an acceptable amount of data load.

Quality for users As already mentioned, different typologies of users (different age bands or origins) imply a careful study on User interface design characteristics. User's interaction should result pleasant and fluid, encouraging people to use IT Kiosk to lighten also Box Office and PDA modules workloads (remembering Box Office and PDA offer some of the functionalities offered also by the Kiosk but they are not characterized by only these ones). **Usability** is therefore considered one of the most important design goals.

Application structure

Identified design goals to achieve, let's continue on defining a design overview of IT Kiosk subsystem considering the following analysis steps:

- Determination of the application typology according to requirements and application constraints
- Identification of deployment constraints
- Choice of the architectural style
- Determination of technologies to deploy

Type of application Among different application typologies proposed in literature (see Architecture Guide 2.0, Beta, Microsoft), Rich Client Application and Web Application resulted the most appropriate to our application context. Both offer a User Interface by what user can interact with the system, but they differ for different deploying modalities and portability, usability and scalability levels. Rich Client application is ideated as a stand-alone application (operating on a single machine) and is suitable for scenarios in what a massive local network communication is needed (DB local server). Suitable for an expert usage, it offers powerful graphical instruments. Web Application offers on the other hand high portability (it can work on different operative systems thanks to the web browser visualization), scalability (no installation is needed to execute different instances) and is suitable for situations that require an almost continuous communication with remote machines. Their graphic simplicity allows a novice user to simply interact with the application. Considering IT Kiosk functional requirements, Web Application resulted the most appropriate. All functionalities require in fact an almost continuous communication with a DB server, deployed in a local network in this case. Keeping in

mind Usability design goals, these web applications offer also a simple and pleasant graphic that allows user to fluidly and quickly interact with the module. High scalability level is finally guaranteed, since to install more than one IT Kiosk module in a network we just need to execute different web browser instances connect to the same DB server, without taking care of the OS characteristics. We propose in fig. 5.11 schema that should better explain the features of the chosen type of application.

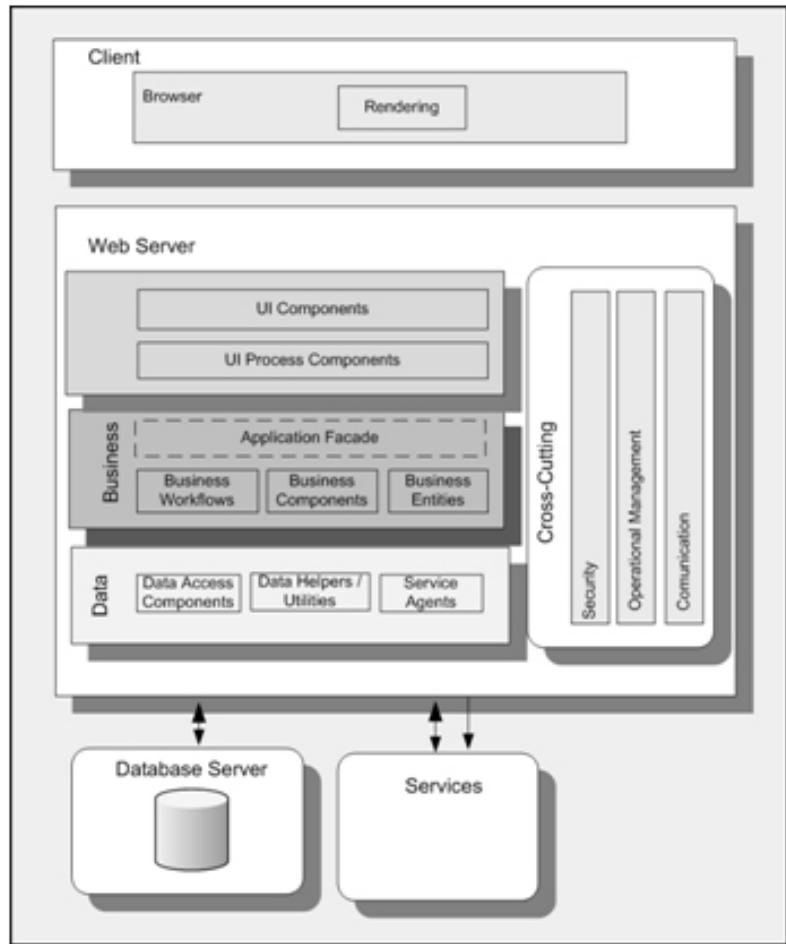


Figure 5.11. II Kiosk application schema

Deployment constraint As already mentioned, IT Kiosk is a component of the IT Beach system whose architecture is a centralized repository type. Infact, every subsystem should communicate with a remote DB to gather the information necessary to execute their functionalities. Considering this deployment constraint, it's necessary keeping in mind that the IT Kiosk subsystem should be provided with a remote communication paradigm for a correct and complete running. Every IT Kiosk module should be deployed on a single machine, and either graphic component and logic core should be installed and

run on the same emplacement. DB remote communication will use a network interface to connect to a local network. The deployment architecture chosen is the not-distributed type shown in fig. 5.12.

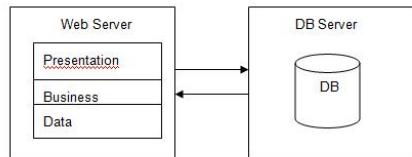


Figure 5.12. IT Kiosk deployment schema

Architectural style Basing on the experience of the design and implementation team, architectural style chosen resulted to be that one named, in literature, “3-Layered”. This architectural style is characterized by a 3 logic levels decomposition that are:

- Presentation, logic level grouping all components related to the rendering of user interfaces and the management of user’s interactions.
- Business logic, logic level whose components are dedicated to manage application logic core in terms of data processing, algorithms execution and user interfaces management(opening, closing).
- Data Access, logic level grouping all components related to manage data entities and local or remote DB access procedures. Layered structure guarantees high Maintainability and extensibility thanks to the decomposition of system elements in different areas of competence. Subsystem results, therefore, easily modifiable and manageable in case of future necessary modifying.

Layered decomposition allows to realize slim, simple system architecture, simply manageable in short and long terms, satisfying in this way the many design goals treated in the previous paragraphs.

Technology selection Strictly connected to the application typology is the technology to deploy. ASP.Net technology resulted to be, after many researches, the most appropriate and advanced regarding web applications development. We’ll therefore utilize .Net framework and ASP.Net technology in conjunction with C# language to develop IT Kiosk sub system components.

Key Hot Spots

Design phase considers many relevant and necessary key points to achieve the previous set design goals. They concern to many quality attributes and a certain number of ”transverse choices” (that involve every layer of the architecture). Starting from the quality attributes key points we find:

- Performance
- Security
- Scalability
- Availability

Transverse choices imply the presence of the following design key points:

- **Authentication and authorization**, it will consist in defining authentication procedure, server and client side, to guarantee that the parts would be authorized to send or receive data, and define an access control mechanism based on user's roles and responsibility definition.
- **Caching and State**, the sub system would try to minimize the number of accesses to DB for static information supplying, like the ones related to services, activities and news; the goal is to reduce the network data load decreasing the DBMS workload. After the user login, the subsystem should also maintain an authentication state for the entire duration of the user interaction till the explicit logout procedure execution.
- **Communication**, considering sensitive data transmitted, related to payments or customers' credentials, every end-to-end communication should be encrypted and authenticated as principal countermeasures to sniffing and spoofing attacks.
- **Concurrency and transactions**, read and write locking DB mechanisms should be implemented to satisfy ACID properties of DB transactions.
- **Exceptions management and logging**, a precise mechanism to raise and log exceptions from lower to higher layers has to be implemented in order to reduce the probability that the system could crash without any notify.
- **Logging**, registering all operational events like login procedures, private area accesses or exceptional events, will allow administrator to check constantly the system and identify accidental faults or attacks directed to security and the correct running of the system.
- **Data Validation**, server and client side to reduce inconsistency states, errors and crash of the sub system.

Candidate solution

The design study and choices completed in the previous paragraphs can be reassumed in the figure 5.13.

Description: Every customer is allowed to access the Kiosk functionalities, with the difference that only the Periodic customer owns a personal data area accessed by an authentication procedure. Every Kiosk software module is characterized by a Layered internal architecture, and communicates with the DB server using an encrypted communication. Every error or system faults raising an exception is instantly logged in one or more text files.

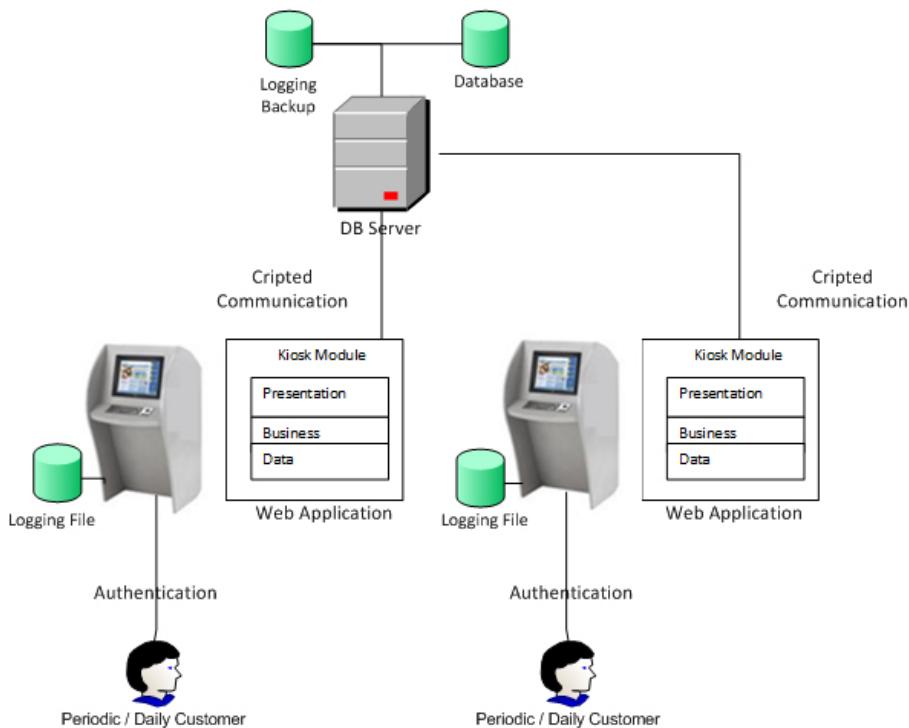


Figure 5.13. Candidate solution for the IT Kiosk

5.3.6 PDA services module

The PDA services module is an important part of the ITBe@ch system because it's the only module that allows to control the right of access of the customers inside the structure. Even though the access to the resort it's controlled by the turnstiles at the entrance, it may happen that one can avoid this control and try to benefit the services of the structure without the right to it: with this module is possible to avoid this problem. We'll analyse now the design issues for this module.

Preliminary design considerations

The module related to the PDA services presents some design considerations that are different from the other modules. Such a difference is due to the fact that the platform where this module will be working is different from the others as the device is not a PC but a mobile device. This is a constraint especially regarding libraries to use, problems linked to the energy saving that every mobile device has to respect due to the short battery life, to the dimension of the screen which being small doesn't allow a complete visualization of forms and windows as it's possible in the other modules that work in a pc using a big LCD screen.

In this paragraph we'll analyse in detail these issues and the solution proposed to solve the problems related to them: we'll take into consideration some design solutions that will allow the module to work properly. Let's summarize these issues below:

1. The PDA services module will be used by a single employee adequately trained; therefore usability can be considered as a not highly critical factor.
2. The PDS services module, working on mobile devices, must communicate with the remote server where the database is hosted through a not wired network. This brings to an increase of the security issues between the device and the server, as the wireless communication can be easily "sniffed". Data exchanged by server and PDA must be therefore encrypted or an encrypted channel for the communication must be used.
3. Another problem related to wireless networks is the fact that they're often discontinuous or not working. Particular attention will be therefore given to the data caching and to messages queuing so that short interruptions of the network service doesn't affect the correct system's behaviour which has to appear as normally working.
4. Despite of the fact that this module communicates with the others in an indirect way throughout the central repository (this makes subsystems highly independent among each other) its design cannot be not linked from the other module's ones, in order to reach a certain homogeneity between the subsystems.
5. The reduced dimensions of the screen makes particularly critical the visualization of the windows and in particular of the written data. Particular attention must therefore be given to the colours used, to the dimension of the font and in general to the readability of the data.
6. The limited duration of the battery raises issues related to the energy saving. We will adopt policies related to the connection to the network which can be activated only when necessary for example, or requests to the DBMS can be reduced in order to get more data at one reading when it's supposed that the request of the other data will be made in a second time.

Design goals

In this paragraph we will list the design goals to achieve for the PDA services module. Starting from those ones already mentioned recruits and constraints related to technologies to deploy and subsystem's infrastructure, we'll end with the quality attributes objectives belonging to three distinct areas: quality in use, design quality and quality for user.

Quality in use When analysing the functionalities that the PDA services module will have to offer, it's easy to notify that they don't result to be extremely critic for the overall system functioning. Infact, considering the two macrofunctionalities, that is to say the services purchasing and the customers control inside the structure, we can see that the first functionality can be anyway take place in the IT Kiosks present inside the strucruta, while the second is a functionality that doesn't have to be applied continuously: the system can continue working a few hours without any control of the customers without anybody notice that. Therefore, differently from the Box Office, the availability of the subsystem we're now analysins is not critic for the overall system. Even though the module doesn't require to be available all the time, its reliability must be anyway taken into consideration: the PDA is going to interact directly with a customer who requires a service or that is going to be checked; for this reason malfunctions that causes failure must be avoided during it's use. Also, situations where the output is inconsistent must be avoided: since the PDA will communicate with the DBMS for the handling of the virtual account of the customer, it will never have to happen that at the end of an operation, due to a not well defined error, the beach assistand who's using the PDA find himself in the condition where he doesn't know the outcome of the operation just ended. As we already said above, particular attention must be given to the security and in particular to the data transmission through the wireless network. First of all, it's necessary to avoid an unauthorized access to the functionality of the subsystem, which will have to be protected by authentication and authorization. Therefore, denial of service and data sniffing of reserved data must be avoided.

Design quality The module has to consider the following quality attributes, putting their achievement among its design goals:

- **Performance.** Performance is an indication of the responsiveness of a system to execute any action within a given interval of time. It can be measured in terms of latency or throughput. Latency is the time taken to respond to any event. Throughput is the number of events that take place within given amount of time.
- **Security.** Security defines the ways that a system is protected from disclosure or loss of information, and the possibility of a successful malicious attack. A secure system aims to protect assets and prevent unauthorized modification of information.

Quality for the user As we already said above, this module will be used only by a well trained user and the interaction with the customers inside the structure will be

minimal. This motivations, together with the limited functionalities it will implement makes usability not a really important issue for the realization of this subsystem.

Application overview

Identified design goals to achieve, let's continue on defining a design overview of IT Kiosk subsystem considering the following analysis steps:

- Determination of the application typology according to requirements and application constraints
- Identification of deployment constraints
- Choice of the architectural style
- Determination of technologies to deploy

Type of application The type of application chosen for this subsystem is "Mobile Application" developed with a "rich client". This type of application results to be particularly useful for systems that operate in mobile devices (such as PDA or smartphones). The fact that this application will be developed as a rich clients means that all the layers will be present in the same device(that is to say the graphical interface and the business layer): the only element out of the device is the repository with the data (the database as we know is located in the database server). We'll now give an overview of how the application will be structured in terms of application deployment, architectural style and technologies used.

Identification of deployment constraints The deployment expected for this module is of non distributed type, that is to say that the components for the interaction with the users and the logic necessary to make it works will be in the same device. As we said above, the data will instead be present in the remote database server: for this reason the device where the module will be deployed will have to contain a wireless module to allow the communication between the device and the server. Besides, the RFID reader used in this module will communicate through SDIO⁵ port: for this reason the PDA used to host this module will have a port compatible with this kind of readers.

As the device will use Windows Mobile as operative system, the deployment will be operated through ActiveSync using a Windows Installer file (MSI).

⁵SDIO stands for Secure Digital Input Output. An SDIO card is a combination of an SD card and an I/O device. This kind of combination is increasingly found in portable electronics devices. Devices that support SDIO (typically PDAs like the Palm Treo, but occasionally laptops or mobile phones) can use small devices designed for the SD form factor, like GPS receivers, Wi-Fi or Bluetooth adapters, modems, Ethernet adapters, Bar code readers, IrDA adapters, FM radio tuners, TV tuners, RFID readers, digital cameras, or other mass storage media such as hard drives.

Architectural style Basing on the experience of the design and implementation team, architectural style chosen resulted to be that one named, in literature, ”‘3-Layered”’. For an explanation of this architectural style refer to the same paragraph for the box office module.

Technology considerations For the development of this module, nor WPF and regular windows forms can be used; for this reason, using the .NET framework used to build the Box Office is not possible. We’ll use the .NET Compact Framework Instead. The .NET Compact Framework is a subset of the .NET Framework class library and also contains classes exclusively designed for it.

Key Hot Spots We’ll list now the design Key Hot Spots for the PDA services module:

- This module will be developed as a Rich Client application
- The windows used must be visible and easily readable even in the presence of a strong light as the sun one on a PDA device using Windows Mobile 6.0 as operative system with a resolution of 640x480 DPI.
- The windoes must be developed taking into consideration the constraint related to the deployment platform: trying to adapt the graphic interface of the other subsystems to this one would be an error. This GUI will have to be designed from scratch taking into consideration all the limitations and constraints typical of PDA devices.
- As the application will be of Rich Client type and all the layers will be present in the same device, in order to optimize the performance and to adopt a ”‘memory saving”’ police will be important to adopt components-reuse policies in order not to repeat code for the same functionalities and to save memory.
- The module will have to deploy ”‘battery saving”’ policies when possible.

5.4 Hardware

We list in the table 5.1 all the hardware components that could be useful to implement ITBeach subsystems.

Quantity	Component	Description
1 x	PC and related peripherals	ITBeach "Box Office module" running machine
3 x	PC and related peripherals + touch screen	ITBeach modules running machine:Bar module, IT Kiosk module
3 x	PDA with SDIO interface	ITBeach "PDA module" running machine
2 x	Thermal 80 mm printer	Used by " Box office module" to print not-fiscal receipts for subscriptions and services purchases
2 x	Mobile thermal 60 mm printer	Used by " PDA module" to print not-fiscal receipts for subscriptions and services purchases
2 x	USB pin pad	Password insertion device to complete RFID tag payment method
5 x	Uninterruptible Power Supply (UPS)	To guarantee power supplying continuity to hardware devices
2 x	Waterproof keypad	Keypad to be deployed in IT Kiosk machine for user interaction
5 x	Desktop USB RFID reader; standard ISO 14443	Desktop RFID reading devices to be used in modules:Box Office, IT Kiosk, Remote Interface, Bar
3 x	Mobile SDIO RFID reader; standard ISO 14443	Desktop reading devices to be used in PDA module
2 x	Turnstile with RFID reader integrated	Accesses control gate by RFID tag reading
1200 x	RFID wearable tags	Necessary tags to use to enable all fucntionalities

Table 5.1. System hardware components

Chapter 6

System development

6.1 Overview

As we already said in the previous chapter, we decided to implement our system using the layered model. Before analysing how we developed each layers, let's introduce what layers are and how they characterize a system. Layers are the logical groupings of the software components that make up the application or service[5]. They help to differentiate between the different kinds of tasks performed by the components, making it easier to create a design that supports **reusability of components**. Each logical layer contains a number of component types grouped into sub-layers, with each sub-layer performing a specific type of task. By identifying the generic types of components that exist in most solutions, it's possible to construct a meaningful map of an application or service, and then use this map as a blueprint for other project's design. Splitting an application into separate layers that have distinct roles and functionalities helps to:

- maximize maintainability of the code;
- optimize the way that the application works when deployed in different ways;
- provide a clear delineation between locations where certain technology or design decisions must be made.

6.1.1 Presentation, business and data services

At an abstract level, the logical architecture view of a system can be considered to be a set of services co-operating among each others grouped into the above described layers, as we can see in figure 6.1¹:

In the figure above we can identify three basic sets of services:

- **Presentation Services:** are the user-oriented services responsible for managing user interaction with the system, and usually consist of components located within

¹Application architecture Guide 2.0

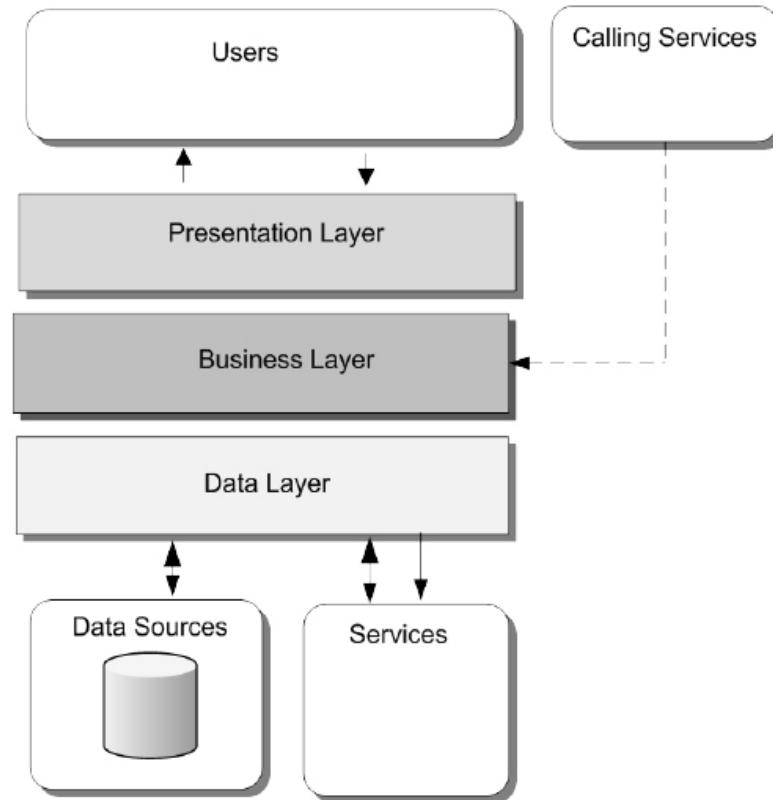


Figure 6.1. The logical architecture view of a layered system

the presentation layer. They provide a common bridge into the core business logic encapsulated in the business services.

- **Business Services:** implement the core functionality of the system, and encapsulate the relevant business logic. They generally consist of components located within the business layer, which may expose service interfaces that other callers can use.
- **Data Services:** provide access to data that is hosted within the boundaries of the system, and data exposed by other back-end systems; perhaps accessed through services. The data layer exposes data to the business layer through generic interfaces designed to be convenient for use by business services.

We'll now describe in a deeply way how we implemented these 3 layers in our project starting with the Data Layer.

6.2 Data layer

6.2.1 Overview

In a N-Tier architecture, the data access layer consists of components that aid one in the process of accessing the database. When it is used correctly, the data access layer serves as an abstract level for the structures of the database. Simple changes that are made to the database and to tables and other components will not affect the rest of the application - the data access layer deal with this issue. The various layers of the application send their data requests to this layer, and this layer responds. Only the data access layer may access the database. As a result, this is where field names and table names are hard coded - they are not hard coded anywhere else in the system. The data access layer is also able to access many services that it can then provide with data - these might include Services and Active Directory. Importantly, the data access layer serves as an extra layer for database security. This is because the other layers do not know database credentials, connect strings, or other sensitive information - with the data access layer, there is simply no reason for them to have access to this knowledge. On the data access layer, one can write generic methods, which will then interface, with custom data. For instance, one might want to write a method for the creation and subsequent opening of a Connection object (internal), plus one more for the creation and use of a Command object, as well as a stored procedure with or without a return value... The possibilities are endless. It can also contain specific methods, so that in the event that the Product object calls on it with the appropriate data, it can then persist it along to the Data Tier. Obviously, the data layer should contain no business rules or data manipulation / transformation logic - that is for other layers. The data access layer just serves as a reusable interface to the database - no more, no less. Anyway, the approach of writing all the possible methods to access the data is long and results to be impossible for large databases. For this reason there are automatic ways to develop this task. We'll follow a path to explain this feature.

First of all we should consider the difference between persistence logic and the data itself. The reasons for separating persistence logic from the data include the following:

- Separate data persistence components can isolate the application from database dependencies, such as the name of the data source, connection information, and field names.
- Many of today's applications use loosely coupled, message-based technologies. These applications typically communicate by passing business documents, rather than by passing objects.

To attain the distinction between persistence logic and the data itself we'll define two different component types.

- Data access logic components. Data access logic components retrieve data from the database and save entity data back to the database. Data access logic components also contain any business logic needed to achieve data-related operations. From now on we'll refer to Data Access Logic Components as DALC.

- Business entity components. Data is used to represent real world business entities, such as clients or tags. There are numerous ways to represent these business entities (e.g.XML, DataSets or custom object-oriented classes) depending on the physical and logical design constraints of the application. In our case we choose to use the third option, but we'll analyse this more in detail later in this chapter.

Data access logic components

A DALC provides methods to perform the following tasks upon a database, on behalf of the caller:

- Create records in the database.
- Read records in the database, and return business entity data to the caller.
- Update records in the database, by using revised business entity data supplied by the caller.
- Delete records in the database.

The methods that perform the preceding tasks are often called "‘CRUD’" methods (we saw the meaning of CRUD in the previous chapter). The DALC also has methods to implement business logic against the database. For example, a DALC might have a method to find the most popular service in the past month. Typically, a DALC accesses a single database and encapsulates the data-related operations for a single table or a group of related tables in the database. For example, one might define one DALC to deal with the Activities and Services tables in a database, and another DALC to deal with the Customers and Subscription tables.

Business entities

Each DALC deals with a specific type of business entity. For example, the Subscription DALC deals with Subscription business entities. There are many different ways to represent business entities, depending on factors such as the following:

- Is the business entity data needed to controls a Windows form or on a web page?
- Are sorting or searching operations on the business entity data?
- Does the application deal with business entities one at a time, or does it typically deal with sets of business entities?
- Will the application be deployed locally or remotely?
- How important are nonfunctional requirements, such as performance, scalability, maintainability, and programming convenience?

The possible implementations are listed below:

- XML. XML Document Object Model (DOM) object can be used to represent business entity data. XML is an open and flexible data representation format that can be used to integrate diverse types of applications.
- DataSet. A DataSet is an in-memory cache of tables, obtained from a relational database or an XML document. A DALC can use a DataSet to represent business entity data retrieved from the database.
- Typed DataSet. A typed DataSet is a class that inherits from the ADO.NET DataSet class and provides strongly typed methods, events, and properties to access the tables and columns in a DataSet.
- Business Entity Component. This is a custom class to represent each type of business entity. Fields are defined to hold the business entity data, and properties to expose this data to the client application. Methods are used to encapsulate simple business logic, making use of the fields defined in the class. This option does not implement CRUD methods as pass-through methods to the underlying DALC; the client application communicates directly with the DALC to perform CRUD operations.
- Business Entity Component with CRUD behaviors: A custom entity class as described previously is defined, and CRUD methods that call the underlying DALC associated with this business entity are implemented.

6.2.2 Mapping relational data to business entities

Databases typically contain many tables, with relationships implemented by primary keys and foreign keys in these tables. When one defines business entities to represent this data in an application, it must be decided how to map these tables to business entities. When business entities are defined to model the information in the database, it should be considered how information in the application will be used. It's therefore important to identify the core business entities that encapsulate application's functionality, rather than defining a separate business entity for each table. For example in our case it could be an idea to create a Tag entity that access also the type of tags persistent tier.

Implementing Data Access Logic Components

A DALC is a stateless class, meaning that all messages exchanged can be interpreted independently. The DALC provides methods for accessing one or more related tables in a single database, or in some instances, multiple databases as in the case of horizontal database partitioning. Typically, the methods in DALC invoke stored procedures to perform their operations. One of the key goals of DALC, as we already saw, is to hide the invocation and format of the database from the calling application. DALC provide an encapsulated data-access service to these applications. Specifically, DALC handle the following implementation details:

- Manage and encapsulate locking schemes

- Handle security and authorization issues appropriately
- Handle transaction issues appropriately
- Perform data paging
- Perform data-dependent routing if required
- Implement a caching strategy if appropriate, for queries of nontransactional data
- Perform data streaming and data serialization

In the case of .Net framework, DALC use ADO.NET to execute SQL statements or call stored procedures. If the application contains multiple data DALC, as it's supposed to be, one can simplify the implementation of DALC classes by using a data access helper component. This component can help manage database connections, execute SQL commands, and cache parameters, operations that would be exactly the same for each DALC class. The DALC still encapsulate the logic required to access the specific business data, whereas the data access helper component centralizes data access API development and data connection configuration, thereby helping to reduce code duplication.

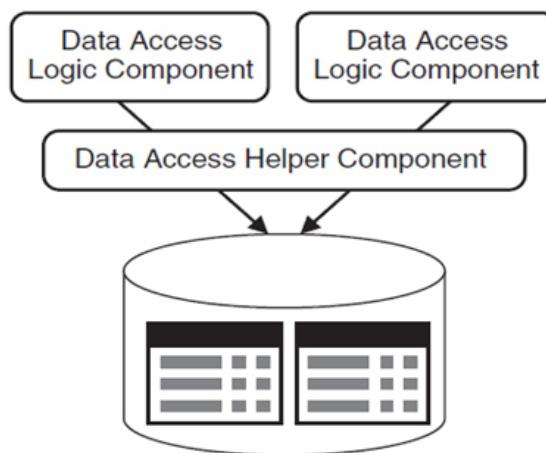


Figure 6.2. Data access helper

If there are utility functions that are common to all of DALC of an application, a base class for DALC to inherit from and extend can be defined.

To support a diverse range of business processes and applications, let's consider the following techniques to pass data to and from DALC methods:

Passing business entity data into methods in DALC.

- Data can be passed in several different formats: as a series of scalar values, as an XML string, as a DataSet, or as a custom Business Entity Component. Returning business entity data from methods in the Data Access Logic Component.

- Data can be returned in several different formats: as output-parameter scalar values, as an XML string, as a DataSet, as a custom Business Entity Component, or as a data reader.

Implementing business entities

Business entities exhibit the following characteristics:

- Business entities provide stateful programmatic access to business data and (in some designs) related functionality.
- Business entities can be built from data that has complex schemas. The data typically originates from multiple related tables in the database.
- Business entity data can be passed as part of the I/O parameters of business processes.
- Business entities can be serializable, to persist the current state of the entities. For example, applications may need to store entity data on a local disk, in a desktop database if the application is working offline, or in a Message Queuing message.
- Business entities do not access the database directly. All database access is provided by the associated Data Access Logic Component.
- Business entities do not initiate any kind of transaction. Transactions are initiated by the application or business process that is using the business entities.

In our application, as we already said, we used the approach to represent data with custom business entities, that is to say custom C# objects. Custom classes that represent business entities typically contain the following members:

- Private fields to cache the business entity's data locally. These fields hold a snapshot of the data in the database at the time the data was retrieved from the database by the Data Access Logic Component.
- Public properties to access the state of the entity, and to access subcollections and hierarchies of data inside the entity. The properties can have the same names as the database column names, but this is not an absolute requirement. Choose property names according to the needs of your application, rather than the names in the database.
- Methods and properties to perform localized processing by using the data in the entity component.
- Events to signal changes to the internal state of the entity component.

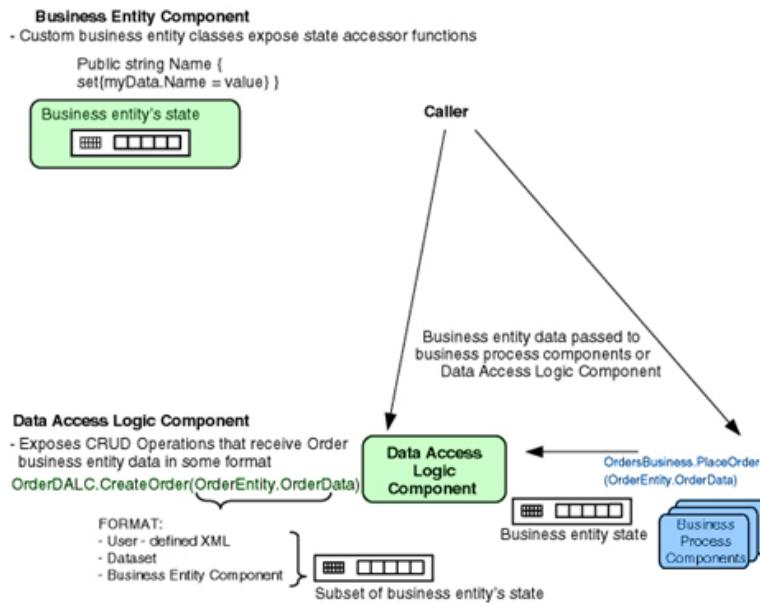


Figure 6.3. Business entities components

The entity class has no knowledge of the DALC or the underlying database; all database access is performed by the DALC to centralize data access policies and business logic. Also, the way in which business entity passes data through the tiers is not directly tied to the format of the business entity representation. Now that we've explained the concepts at the base of the data layer development, let's see what Object Relational mapping technique is and how this helped us in the development of our project.

6.2.3 Object-relational mapping

Object-relational mapping (ORM) is a mechanism that makes it possible to address, access and manipulate objects without having to consider how those objects relate to their data sources. ORM lets programmers maintain a consistent view of objects over time, even as the sources that deliver them, the sinks that receive them and the applications that access them change. Based on abstraction, ORM manages the mapping details between a set of objects and underlying relational databases, XML repositories or other data sources and sinks, while simultaneously hiding the often changing details of related interfaces from developers and the code they create. ORM hides and encapsulates change in the data source itself, so that when data sources or their APIs change, only ORM needs to change to keep up—not the applications that use ORM to insulate themselves from this kind of effort. This capacity lets developers take advantage of new classes as they become available and also makes it easy to extend ORM-based applications. In many cases, ORM changes can incorporate new technology and capability without requiring changes to the code for related applications. Data management tasks in object-oriented (OO)

programming are typically implemented by manipulating objects that are almost always non-scalar values. For example, consider an address book entry that represents a single person along with zero or more phone numbers and zero or more addresses. This could be modeled in an object-oriented implementation by a “person object” with “slots” to hold the data that comprise the entry: the person’s name, a list (or array) of phone numbers, and a list of addresses. The list of phone numbers would itself contain “phone number objects” and so on. The address book entry is treated as a single value by the programming language (it can be referenced by a single variable, for instance). Various methods can be associated with the object, such as a method to return the preferred phone number, the home address, and so on. However, many popular database products such as structured query language database management systems (SQL DBMS) can only store and manipulate scalar values such as integers and strings organized within normalized tables. The programmer must either convert the object values into groups of simpler values for storage in the database (and convert them back upon retrieval), or only use simple scalar values within the program. Object-relational mapping is used to implement the first approach. The heart of the problem is translating those objects to forms that can be stored in the database for easy retrieval, while preserving the properties of the objects and their relationships; these objects are then said to be persistent. As we saw before, this mechanism is easily manageable with ORM techniques.

Automatic ORM tools: why to use them

Whether developing a small or a big application, almost always one have to deal with data. It’s even a critical part of an application. Problem is this is tedious, repetitive work, which consumes a lot of the time we would prefer to spend on other parts of the application. Without forgetting that the less interesting the work is, the higher the risks of errors. Automatic tools for generating ORM exists for this reason. Their goal is to simplify the creation of data access layers, automate data access, or generate data access code.

As we said above, the principle of object-relational mapping is to delegate to tools the management of persistence, and to work at code-level with objects representing a domain model, and not with data structures in the same format as the relational database. Object-relational mapping tools establish a bidirectional link with data in a relational database and objects in code, based on a configuration and by executing SQL queries (dynamic most of the time) on the database.

In terms of tools, the offer is huge. This is true for .NET as well as for Java. There is anyway an impressive quantity of tools for both sides. The most important when searching for the right tool is to define precisely which criteria are essential for the application that is going to be developed and their development criteria.

Let’s analyze some of the most important criteria used to choose whether to adopt a automatic ORM.

Basic features

- Be able to use inheritance, create hierarchies between entities, and use *

type of relations

- Support for transactions
- Aggregates (equivalent to SQL's SUM, AVG, MIN, MAX, COUNT)
- Support for grouping (SQL's GROUP BY)

Useful extended features

- Supported databases (e.g. SQL server, mySQL, Oracle).
- Support for DataBinding (to be able to bind data objects to visual components). Note some specificity exists with Windows Forms.

Flexibility

- Customization of queries. We often need to go beyond what is possible with the provided query language. In these cases, we need to be able to provide custom SQL queries.
- Support any type of SQL joins (inner join, outer join)
- Concurrency management (support for optimistic and pessimistic approaches)
- Support for the data types specific to the database management system (identity columns, sequences, GUIDs, autoincrements)
- Be able to map a single object to data coming from multiple tables (joins, views). Most of the tools handle a direct mapping of a class to one table. We often need more.
- Be able to dispatch the data from a single table to multiple objects.
- Generation of the database schema.

Evolution, compatibility

- Maintainability
- Possibility to move to a new mapping tool
- Serialization. Serialization can be used to persist data outside of the database. Serialization can be done into a binary format, or more important, in XML
- Distributed objects

6.2.4 .netTiers

.netTiers is a set of open source code generation templates based on CodeSmith that simplify the tasks of creating customized Application Tiers for Microsoft .Net applications. Generates a fully compilable solution along with separate projects and framework tiers for your application. Here's a list of the most important features of this tool:

- Creates a full suite of stored procedures that are specialized towards application domain. This code can run inline, as parameterized sql as well and does not have to be used as procedures.
- Automatically generates entity objects and their relationship objects as a domain based on the tables of the database.
- Advanced Entity Validation Rule Engine, which can use any of the pre-created or use delegates for any custom rules.
- Classes include both partial classes and concrete classes throughout so that it's possible to customize logic that will not be overwritten.
- Uses a custom generic List for collections that supports all .net ComponentModel interfaces and more, and are Bindable, Sortable, and Filterable.
- Creates a full set of typed DataSource controls for your entire API with design time support, they are similar to the ObjectDataSource, only these are full featured and are actually developer friendly.
- Code is fully commented and follows Microsoft naming guidelines throughout.
- The Data API consists of query support by Primary Key, Foreign Key, Indexes, ManyToMany join relationships, All, Paged Select with Filter, and Find, along with database write methods such as Insert/Update/Delete.

How does .netTiers figure out what to generate? .netTiers uses the rich meta data CodeSmith provides that is found from database in order to create a complete object domain based on data relationships. The SchemaExplorer provides information on Database, Table, View, Indexes, StoredProcedures, and more. There are several important aspects to keep in mind when creating or using an existing database.

Now let's see how .netTiers deal with the previous explained concepts of Business Entities and DALC. First of all let's try to understand how the code generated with this is used in our project. As we already explained in the design part, the programming language we used to develop each module was C#, that is to say .NET Framework. For this reason, the data layer must be compatible with this platform. What .netTiers produce is a list of dll libraries. The business layer is therefore accessing the database through these libraries, and doesn't know anything of how the access is performed. This means that even if the database is moved to another location, even if the authentication to access the server is changed, as far as the names of the tables remains the same, the application will continue

to work and no changes will be necessary. .netTiers creates two different libraries for the entities and for the DALC. Let's analyze therefore how these two layers are handled.

Entity layer

An Entity is not a regular .net type, it is a significant type that holds an intrinsic meaning that is essential in your application. Entities can be real or conceptual and in business applications, which .netTiers is tailored to, holds a special meaning to the data. For example: an Employee, an Order and OrderDetail are all entities. Entities are malleable and have a lifecycle, and their types must take extra special checks and balances to ensure the integrity of the entity.

Entity can be seen as the Reflection of a Database-table exposed in Object Oriented fashion. The Word Entity is the same which is commonly used and mostly for a Database-Table. So in the .Net environment an object reflecting a Database-table cascadingly, Exposing its Columns as class-Properties, Carefully matching the Data Types between a database-column and a .Net-Type(Let say string for varchar and Int16 for smallint).

Now think of a Database-Table 'USERS' where, definitely, each row of table is representing a single User's complete information(like her ID, Name, Age and Address) and on the other hand in .Net imagine an Object 'User' (with properties ID, Name, Age and Address). It actually is representing One User from database-table 'Users' or One Row of database-table.

.netTiers' architecture provides us an environment where we can Control the database entities by passing in these .Net entities in special methods. In fact those special methods which Control these .Net entities are called Controller Objects. We'll be studying about them in next articles. for now we should make sure that we know that we can call some dedicated methods from Controller objects to Retrieve an Entity or List of Entities. We can pass in an Entity into Controller behaviors to perform CRUD actions. .netTiers uses the notion of the entity along with the TableModule and Data Transfer Object (DTO) Patterns in order to expose your database as entities. Meaning, for every table in your database, an entity will be generated for that table. The DTO allows you to pass the lightweight entities through the many tiers while still maintaining the loosely coupled open ended architecture of the Entity Layer, since it doesn't depend on any DataProvider.

.netTiers will also attempt to discover all of the relationships that your table has with other tables in the database and will create child properties of those relationships. This will build out your entire entity domain. Currently the relationship types supported are one to one, one to many, and many to many relationships. These relationships make it easy to intuitively work with entities and now have a logical object graph.

Data layer

The data layer in .netTiers is based in the concept of *Repository*. The DataRepository is the entry point into data access API using the default configured data provider. It is the central mechanism for instantiating and loading the individual providers at runtime, and more importantly, retrieving and saving data for data API. The DataRepository itself is

essentially a singleton object using the Decorator Pattern of the individual TableModule Entity Providers. This class lives in the abstract Data Access Layer as mentioned earlier so that it does not reference any particular provider implementation, which only knows about all of entities, their access methods.

Some typical method call looks like:

- DataRepository.BeachesProvider.GetAll();-to get a list of all the beach rows from the database; each row is represented as an entity object.
- DataRepository.BeachesProvider.GetByLocation('Mondello');- get a list of entity (or a single one) from the passed attribute.
- DataRepository.BeachesProvider.Insert(transactionManager, beach);- insert a new row in the database from a code-build entity.

How does the DataRepository know which provider to use? When the DataRepository is first called, it will check the current configuration in the app/web.config in the netTiersService configSection to determine the current default provider and possibly any other provider configurations that have been configured. Configuration was covered in the Getting Started Portion of the documentation. During this process, the data repository will load the default NetTiersProvider, by default the .netTiers provider will be the SqlNetTiersProvider. This class lives in the DataAccessLayer.SqlClient layer of the default generation process.

What are access methods? Access Methods are the methods that comprise the Data API. Depending on how .netTiers is configured it will identify and create a logical set of data access methods utilizing the data model's relationships and how their behavior affects your normalized relationships. That statement is a bit deep, so let's clarify. During the .netTiers generation process, the .netTiers templates will identify and create your initial API methods through based on indexes, keys, stored procedures, and a few other mechanisms all discussed below.

Read Methods:

- Get All
- Get By Primary Key Id
- Get By composite foreign key columns
- Get By Composite index column
- Get By Dynamic WhereClause (Paged Result)
- Get By Custom Stored Procedure

Write Methods:

- Insert

- Update
- Delete
- Save
- Custom - using Custom Stored Procedure

6.2.5 Step-to-step data layer creation

Now that all the concepts of how data layer should be handled, let's see how we proceeded to implement it in our project. In order to be pragmatic, we decided to describe it as a step to step procedure that brought us from the idea of the application to the effective .netTiers library development.

Step 1

Starting from the class diagram we've developed in the analysis phase, we identified the entity objects and created a first model that would include data object used in the system. This is usually not complete, but it gives a clear idea of which kind of information the application will manage. Usually one object corresponds to a table in the database, and its attributes are the same as the attributes of the table. This should not be a surprise, what happens in the development of the data layer for the creation of the library is exactly the opposite as what happens here.

Step 2

Starting from the entity objects diagram, we created the preliminary entity-relationship model. In software engineering, an Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.[4]. The following step was to refine the entity-relationship model by adding new entities and new relationships between entities. Usually at the beginning it's hard to have a complete view of which data will be treated by the system. Obviously, the more the analysis is refined, the more the preliminary entity-relationship module will be close to the final version. But it's not rare that the database structure is continuously modified even when the application is being developed, as writing code raise some issues that were really hard to discover at design level.

Step 3

The next step was to choose which DBMS to use and therefore the physical installation of the database. Since we're using the .NET framework as development platform, the most obvious choice was SQL Server, as it's fully compatible with C#. The version we used for the development was SQL SERVER EXPRESS 2005, which resulted to be complete for

our aims. We had to configure one of our laptop as a temporary server, and in this way we could work with one database and try the locking and concurrency issues.

Step 4

Once the database was installed in the SQL Server, the last step was to create the libraries to be used in the application. After a research in literature of the possibilities to create effective data layers, as we already described in this paragraph we decided to try to use the .netTiers class library generator. The configuration tool allows to connect directly to the SQL Server instance installed in the computer where the tool is running and to select a database from which to create the schema and the class library, as we see in the figure below.

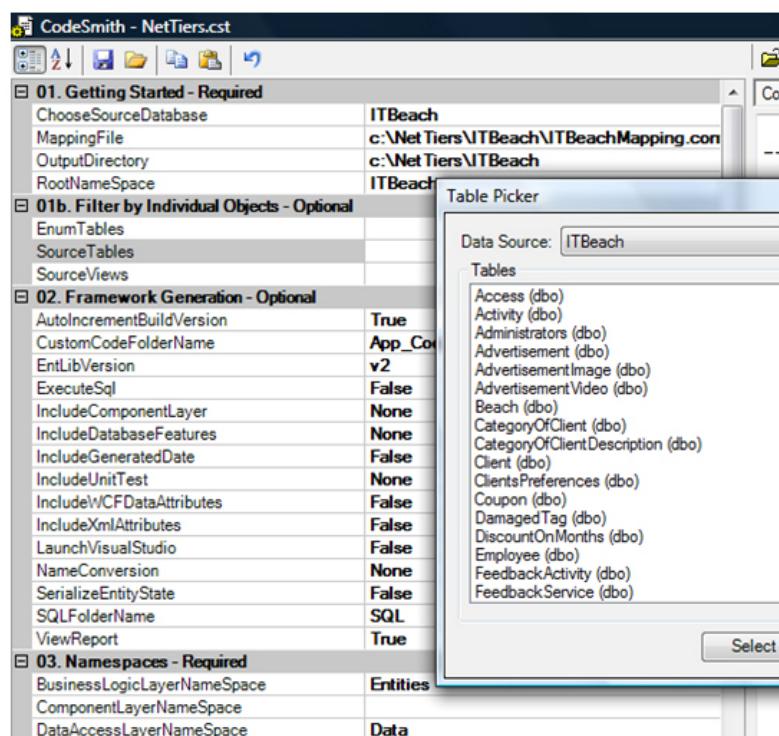


Figure 6.4. Configuration windows of .netTiers for the ITBeach database

After setting the desired attributes and generating the data layer, the tool shows a webpage with the result of the generation if successful; otherwise it shows a list of the errors that didn't make possible the generation. The result of the generation is a .NET solution with 4 class libraries projects based on C# language. The fact that it doesn't generate the libraries directly is due to the fact that one could want to modify the code generated for example to create custom methods or to give some more intelligence to the data layer. In our project we preferred to maintain the data layer as a pure connection with the database and to give all the intelligence to the business layer. So, as we explained

in the paragraph related to the functioning of .netTiers, the result of this path is a library that contains a list of entities, one for each database table and a library which contains the datarepository methods used to access the entities and to operate the CRUD functionalities over them.

A windows project, in order to use the selected library, must use the “app.config” file to manage useful information such as connection to the database. For example, we list the config file for the Remote Interface module:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>

    <configSections>

        <section name="netTiersService"
            type="ITBeach.Data.Bases.NetTiersServiceSection, ITBeach.Data"
            allowDefinition="MachineToApplication"
            restartOnExternalChanges="true" />

    </configSections>

    <connectionStrings>
        <add name="netTiersConnectionString"
            connectionString="Data Source=(local)\SQLEXPRESS;
                Initial Catalog= ITBeach; Integrated Security=SSPI; "/>
    </connectionStrings>

    <netTiersService defaultProvider="SqlNetTiersProvider">
        <providers>

            <add
                name="SqlNetTiersProvider"
                type="ITBeach.Data.SqlClient.SqlNetTiersProvider, ITBeach.Data.SqlClient"
                connectionStringName="netTiersConnectionString"
                providerInvariantName="System.Data.SqlClient"
                entityFactoryType="ITBeach.Entities.EntityFactory"
                useEntityFactory="true"
                enableEntityTracking="false"
                enableMethodAuthorization="false"
                use.StoredProcedure="false"
            />

        </providers>
    </netTiersService>

```

```
</configuration>
```

As we can see in the code listed above, the configuration file is easily manageable: it's possible to change the connection string in case the server is moved to another machine by changing a string, and it's possible to manage various settings such as method authorization and stored procedure use.

The steps 3 and 4 were repeated as many times as necessary because of the modification of the database structure. How these functionalities are accessed by the other layers will be explained later in this chapter.

6.3 Business layer

6.3.1 Overview

This layer contains all the components in charge of managing the core functionalities of a software application. They provide services/functions based on data processing, complex algorithm execution and workflows management, creating therefore a communication bridge between Presentation and Data Layers. It is indeed designed to satisfy Presentation layer data needs, being usually developed on specific requests coming from this above layer development responsibilities. Their tasks are correctly executed retrieving data from the below one, the Data access layer. Considering different tasks they can execute, we can divide business layer components in the following types:

- Business unit components. These components implement the business logic units of the application. They offer simple task services constituted of few data processing steps. Examples of services they provide are CRUD database operations also implying simple processing or reading/writing hardware interfacing.
- Business workflows. These components define and coordinate long-running, multi-step business tasks that can be implemented using also one or more business components. Their tasks could require multiple data processing operations involving database CRUD operations, algorithms execution and hardware interfacing can be combined in any order to obtain a specific result. The Box Office functionality related to the new subscription procedure is a perfect example.
- Business entity components. They represent the real-world business entities they are used to pass data through different layers. They can be organized in specific data objects like DataReader, DataSet, XML streams, Lists but they can also be implemented using custom object-oriented classes. For our application we use object-oriented business entities directly mapped onto database tables to easily manage data.

6.3.2 Business Layer Implementation Approach

Designing a Business Layer(BL) means taking into account the design requirements for the main constituents of the layer, such as business components, business entities and business

workflow components. Different designing steps have to be followed before obtaining all that elements necessary to guarantee the system a reusable, scalable, maintainable and efficient business layer. Thus, designer should perform the following key activities in each of these areas when designing this layer:

- Create an overall design for the business layer, before indentifying and define the singular elements, different design best practices and strategies have to be persecuted. Designer has to determine and analyze security and validation requirements, and identify the appropriate caching and exception management strategies.
- Design business components, designer has to identify the singular business components choosing a particular work division strategy(e.g. creating different components basing on high level functionalities or more grained tasks). Once identified, decisions about their location(e.g. different packages) , coupling and interactions has to be made.
- Design business entity components, indentifying and choose the data format for business objects. They can be realized as DataSet, DataReader, XML stream or they can also be implemented using custom object-oriented classes.
- Design workflow components, Analyzing scenarios and the more complex functionalities, designer can create workflow components by combining new little tasks or ready business components services in order to obtain execute long-running, multi-step business tasks.

In the next paragraph we'll formalize choices we made for the overall design definition, while we'll explain characteristics and issues faced in defining the logic components in the paragraphs dedicated to the implementation details of every module.

6.3.3 Overall Business layer structure

There are several design key hotspots and related issues that have to be considered during Business Layer(BL) overall structure design. These consideration forms the solid base for the next identification and definition of business and workflow components. Following the previous step-by-step design methodology, we start our design process making choices over the BL design key hot spots listed below:

- Caching, Coupling and cohesion, Concurrency and Transactions, Data Access, Exception Management, Logging. Making correct design choices on the previous areas will help us in realizing a high quality layer.

Caching

Designing an appropriate caching strategy for our business layer is important for the performance and responsiveness of our application. Using caching allows to optimize data references, avoid network round trips, and unnecessary and duplicated processing. Particular attention should be also posed on when and how to load the cache data. We

chose to persecute a caching strategy in many of our controllers that are in charge of managing static data. These are data frequently used but rarely changed. We stored them in local variables loaded at application starting to be read many times after. For example, the "New subscription procedure" requires the retrieval of data related to the beach, type of subscription, type of subscription discounts, locations details, customer's category and preferences to register correctly a new customer's subscription. All these information can be loaded only during application starting in order to be visualized immediately. Particular attention has to be posed on the rare modifying of these data. Every time a modification is done, caching updates have to be executed in order to avoid inconsistent situations.

Coupling and cohesion

When designing components, we ensure that they are highly cohesive, and implement loose coupling between layers. This helps to improve the scalability of our system. We chose to create different controllers for every significant functionality for each module. Every module owns its own business components package, sharing other ones that provide common functionalities(e.g. RFID reader interfacing, common data retrieval methods etc.) . We avoided to mix business and data access logic, since the data accesses are executed only by invoking services offered by Data Layer components. This helps to obtain loose coupling between layers.

Concurrency and transactions

The possibility of deploying one or more Box Office(or IT Kiosk) modules lead a concurrent data access system. Locking mechanism have to be deployed to avoid inconsistent situations. Considering that the new subscription procedure of the Box Office module allows to book a particular beach emplacement for a certain period, contemporary reservations on the same location will be avoided only by using a concurrency model. We decided to use the pessimistic concurrency model. We used transaction paradigm when multiple data operations had to be executed to complete a single task. Considering again the creation of a new subscription, its completion implies several writing operations on different tables(e.g. registering customer, subscription, RFID tags, preferences etc.). The failure of one of them, means the failure of the entire task(the creation of a subscription in this case) causing an eventual inconsistent status(e.g. tags and/or customers, or subscription details are not registered). All this tasks were therefore managed by the transaction paradigm that allows to execute a rollback, also when a single failure occurs. Our Data Layer offers an efficient transaction management mechanism. CRUD methods offered by the DataRepository class can be executed in Transaction mode.

Exception Management

A well-designed set of error handling code blocks can make a program more robust and less prone to crashing because the application handles such errors. Raising and handling exceptions is an expensive operation but necessary for realizing a high quality system.

We chose therefore to manage every exception, through the appropriate C# techniques, raising them up to the above Presentation layer.

Logging

Building detection and reaction mechanisms that signal errors or attack conditions into the applications empowers to diagnose and fix problems much more easily. Logging mechanism allow us to reach these objectives. Designing a good logging mechanism is important for the security and reliability of our application. We chose to use the "Logging Application Block" that provides a simple way for our code to log information in a consistent manner across the managed application. Powerful configuration files allow the filtering, routing and formatting of those "Log Entries" to be determined at run time (rather than at compile time).

6.4 Presentation layer

6.4.1 Overview

The presentation layer represents the interface between the user and the other layers of the application. Since the chosen architecture is Strict Layering, this layer communicates only with the user on one side and with the Business Layer on the other. In other words no direct communication with the Data Layer is allowed. Several issues have to be considered while creating the presentation layer. The most important is for sure user experience. The most powerful software, the most complete system will fail to achieve its goals if the user is not happy with the interaction; a lot of errors will be made and this could bring to unexpected states in the layers underneath the presentation one or more critically on the database.

On the other side, one of the most important features affecting user experience is responsiveness, intended as the degree of readiness with which the system gives a feedback to the user after an input is given by him/her. This brief consideration aims to show how the Presentation Layer constitutes a key point for system correct usage and functioning. All the efforts spent in developing a bug-free and powerful system (just an hypothesis since it's impossible to create a bug-free system) would be lost without considering that a human being will have to pilot it. At present days, a broad range of patterns, guidelines and principles exist for Presentation Layer optimization. A group of them concerns the user end interface and another one contains indications for a correct and effective inner structure arrangement and communication establishment with the Business Layer. We'll present these two groups into separate paragraphs (user end issues and architectural issues), provided that many choices made for one side will affect the other, as stated at the start of this paragraph. It's important to clarify that the choices explained in these paragraph concern the whole system as they involves all the subsystems. Module specific considerations will be made later in this chapter, when an overview of each module will be provided.

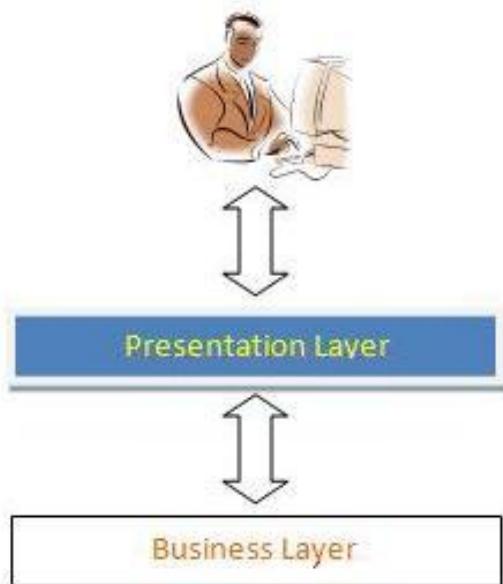


Figure 6.5. Inter layers communication schema

6.4.2 User end issues

The Human-Computer Interaction (HCI) is a study field that from the second part of the last century is continuously evolving as the technology itself evolves providing new devices, systems and interface means. The most recurrent terms are such as usability, ergonomics, communication. All these concepts suppose the existence of a system (not intended as a technological system) made of interacting parts among whose one or more human beings are involved as parts of the system. The most fascinating aspects of HCI are probably the ones involving psychological considerations on how human brain works, processes information and decodes messages. But these considerations would fall far from the scope of this paper and will be therefore omitted.

Usability

Usability is defined as follows: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." (ISO9241-11, 1998) While the first two parameters are quite easy to measure by known metrics like mean execution time, number of errors and so on, the last component of usability is the hardest to monitor, requiring often the submission of questionnaires to users. With the aim of reaching the highest results in all of these three parameters, the presentation layer is been developed keeping in mind a set of usability principles: redundancy of controls, feedback to user (both visual and acoustic), errors notification, graphical coherence, and many more. We propose here the ten usability principles by Jackob Nielsen, the Danish guru of user interface usability:

Visibility of system status: the system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world: the system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom: users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards: users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention: even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

Recognition rather than recall: minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use: accelerators – unseen by the novice user – may often

speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design: dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors: error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation: even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

These ten principles can be satisfied through the use of several techniques and the complete listing of them would require too much words. So the reader should just keep in mind that the user interface has been designed in order to comply with the mentioned principles.

6.4.3 Architecture issues

The considerations made in the previous paragraph affect User Interface Components, that is those components providing a way for users to interact with the application. They render and format data for users. They also acquire and validate data input by the user. but these are not the only components within the Presentation Layer. Aside to them another kind of component could be useful and worth inserting: User Process Components. User process components synchronize and orchestrate user interactions. Separate user process components may be useful in case of a complicated user interface. Implementing common user interaction patterns as separate user process components allows you to reuse them in multiple user interfaces.

The philosophy underneath these components is the separation between merely graphical components and the amount of logic required to orchestrate those components. This logic doesn't have to be confused with the one hosted by the Business Layer. In the first case we're talking about the logic required to manage for example a sequence of views in a multi step task, to dynamically update the information shown by a particular view or to concentrate validation procedures useful to more than one User Interface Component. Business Layer is responsible instead of the domain specific logic (like algorithms and procedures), communication with Data Access Layer to retrieve data and hardware interfacing.

By separating User Interface Components from the Interface logic guarantees a higher level of maintainability and flexibility. The graphical components (like forms, buttons, grids) can be easily substituted with other type of equipollent controls without interfere with the general logic that leads the interaction. When a programmer will be asked to substitute the graphical interface of any of the module composing the system, a robust presentation layer will prevent any pitfall that could derive from a different graphical setting.

The implementation of a user interface and the required presentation logic is definitely

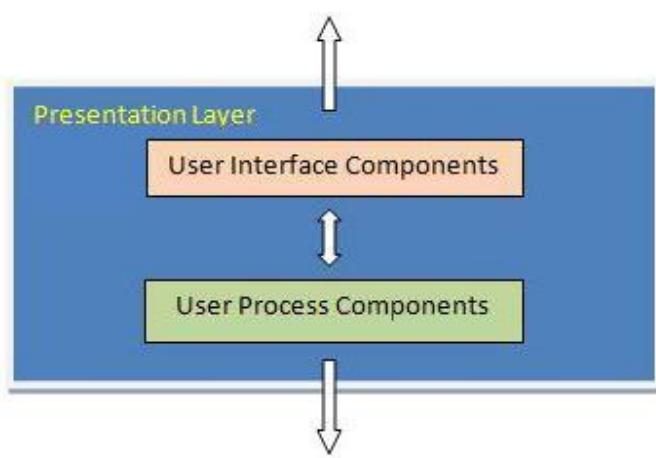


Figure 6.6. Presentation components

different among modules since several features have to be considered:

- Interface devices
- Class of users
- Usage Environment
- Mean length of interaction
- Expectation posed on interaction results
- ... and many others....

That's why module-specific considerations on Presentation Layer implementation will be made in the relative paragraphs in the proceedings of this chapter.

6.5 System decomposition

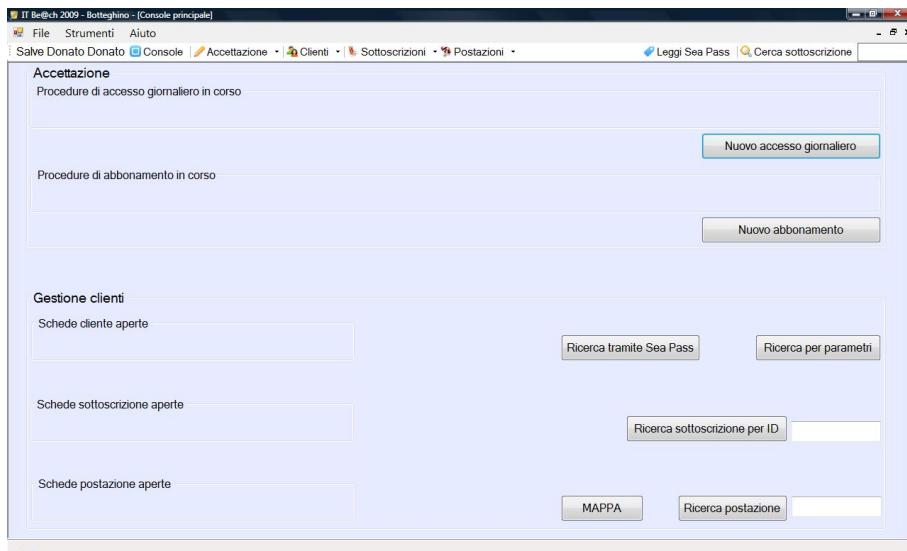
6.5.1 Box office

Introduction

The Box Office Module is the subsystem intended to automate all check in and clients management procedures. These procedures take place in a wooden house just outside the main entrance of the beach resort. Clients interact with the employee through a window on the side facing the street. The employee has to be able to show to the client the locations available for the access with the requested features (type of location, type of subscription, period, etc). As quick as possible, the employee has to provide the client with an RFID tag and a receipt indicating the number of the rented location. The need for a printed receipt arose from the decision by MIIB to avoid providing the beach assistants with an RFID enabled PDA able to discover the rented location associated with the read tag. The type of payment to support is only cash. The box office is also in charge of receiving back the tags from the clients when they're leaving the structure. All the procedures relative to clients management and not available at the IT kiosks will most likely take place in the box office.

Interaction organization

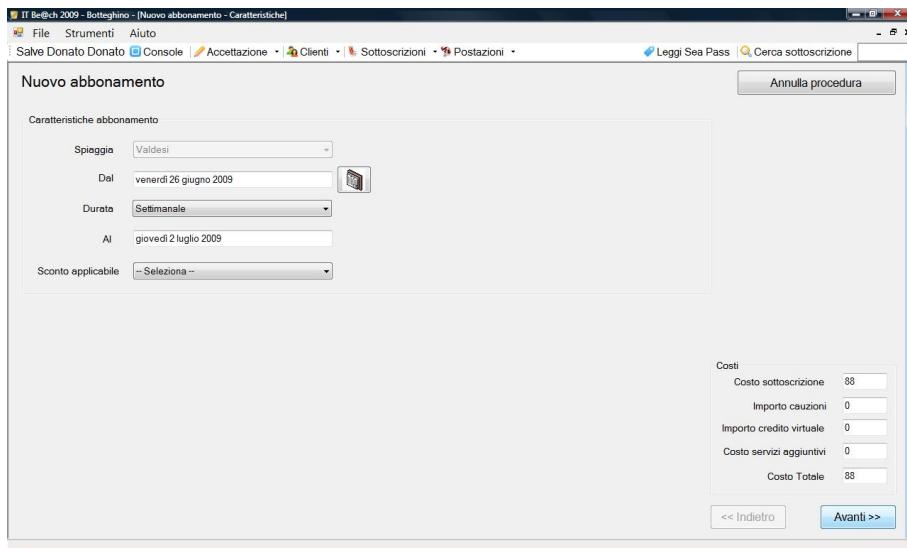
Once the Box Office module is started, it makes a check of database connectivity, loads non transient data on cache, performs some runtime settings and checks for the presence of the RFID reader connected to the computer. If the reader is not detected, the application informs the user of this fact explaining that the functionalities that require tag reading will be disabled. If no problems occur on startup (like no connectivity with the database) the user is requested to insert his/her personal credentials (username and password). Missing to login will bring the application to keep available only the controls needed to close it.



Otherwise the Console Form will be shown and the controls on the menu bar and tool bar will be enabled.

At this point of the interaction, the user may decide to perform the check in of a daily or periodic customer. Let's see the New Subscription procedure first:

1. User starts the task through the relative button on the Console Form or through the toolStripItem on the tool bar. The view relative to the first step of new subscription is shown



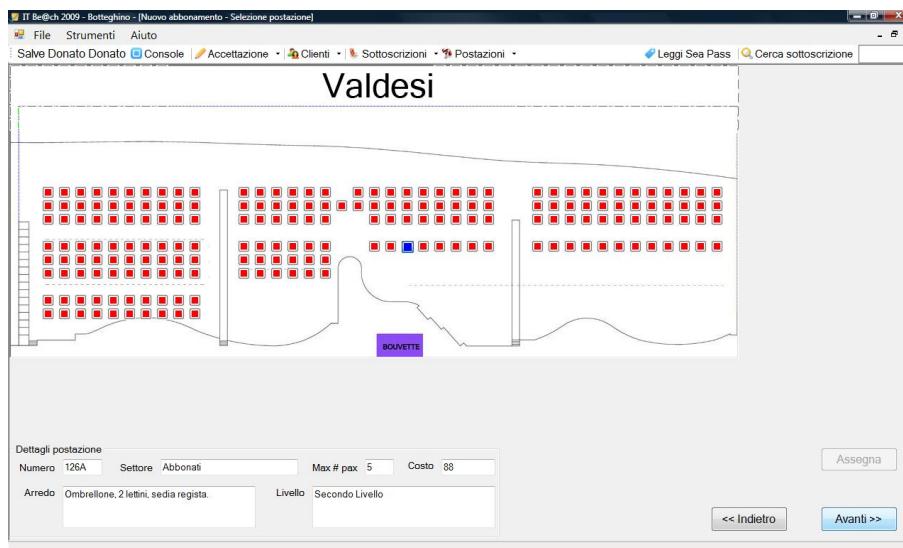
User selects subscription features through the relative controls. An alternative via-keyboard interaction is available, which speeds up the completion of this step. Once subscription features are set, the user may pass to the next step and the Map View

is shown.

2. The Map View will show the locations in the beach resort using three different colors:

- Green: indicating that the location is compliant with subscription features and thus available;
- Red: indicating that the location is busy in one or more days within the selected period;
- Gray: indicating that the location is not available (broken) or not compliant with the selected subscription features (i.e. the first three rows of location could not be available if the user sets a particular type of discount on subscription);

When a green location is selected, the assignation button becomes enabled and gets focus. Even at this step, the user can alternatively use the keyboard but only after having selected an available location of the map with a mouse click: two consequent "Enter" or "Space" presses will bring the user to the next step. When a location is selected its color gets blue.



3. At step 3 the user is asked to insert clients' personal details (name, surname, category of client and so on). During data insertion an RFID tag can be assigned to each client and a virtual account can be activated charging whatever amount the client prefers. The PIN associated with a virtual account can be changed, in order to secure client's money with a code that he/she only knows. A set of 10 preferences (interests) is also selectable in order to collect more information on every client and provide a more customized service to them. Every location needs a minimum number of clients (depending on the type of location: solarium, two sun beds location, . . .). Once this minimum number of inserted client is reached, the user may decide to pass to the last step of the procedure or, alternatively, to insert more clients but anyway no more than the maximum number allowed by the selected type of location.

4. When he press "Next" button, the Step 5 (summary) is displayed.

5. At the Step 5 the user will be able to visualize an overview of all the features selected during the previous 4 steps. If the features satisfy clients' needs, the user may proceed to subscription confirmation and a receipt is printed in A4 format. A Reprintbutton is available in case the user needs to ask another print of the receipt. Note: the receipt reported in figure doesn't correspond to the procedure captured in the previous screenshots but should just be considered as an example of the receipt structure.

The Daily Access procedure is quite similar to the New Subscription one, so we'll show here just an overview of the main differences:

- The first difference is in the Map View, in which it's possible to select more than one location per procedure. This is due to our will to enable the processing of big groups of persons asking for a daily access to the beach resort;
- The next step, the one asking for clients insertion, is slightly different, since the amount of personal data required is reduced. If more than one location were selected at the previous step, at every client insertion user must select one of the locations intended for him/her.
- At the final step, since more than one location could have been selected and filled up with clients, the bill can be partial, by selecting a subset of locations, or total.
- The receipt printed at the end of the procedure is no more in A4 format but targeted on a 80mm thermal printer.

As the application stores the information about check in procedures into the database (in terms of Clients, Subscriptions, Tags,...) it is possible to visualize and eventually modify these information once the desired client/subscription has been identified and the relative management form displayed. In order to perform this identification and visualization there are several ways available:

- The simplest and quickest is probably by reading client's tag: a couple of controls (on Console Form and Parent Form's tool bar) enables tag reading and, if a client is correctly associated to it, the management form is displayed. Having access to a single client will bring to the visualization of all the active subscriptions made by him/her.
- Another chance is to try a parametric search on clients stored on database.

Nome	Data di nascita	Indirizzo
GIOVANNI ARMANIO	06/06/2001	VIA GERARDO ASTORINO 8, 90100, PALERMO, PA, ITALIA
GIOVANNI NOBILE	1/1/1901	DBP, 90100, PALERMO, PA, ITALIA

The search could be based on client's ID (primary key on Clients table) or more naturally on some personal data (i.e. name and surname). If the searched client is

among the ones presented as search result, the user can simply pass to its management form visualization. As for the first case, since the user gets access to a single client, all the active subscriptions made by him/her will be shown.

- Still the user could want to search a specific subscription. In this case the search can be performed by Subscription ID (reported on subscription's receipt). In this case just that specific subscription is shown even if the clients involved have more active subscriptions in that moment.
- As fourth chance, the user may want to visualize the situation for a particular location. From the Console Form is thus possible to start a search based on the location number. If a location with that number exists, the management form shown reports all the active subscriptions for that location.

As a result of any of the preceding procedures, the application will show the management form for the searched element (client, location or subscription).

From this view a series of functionalities are accessible. Proceeding clockwise from the upright corner of the management form:

Location management: user may decide to change the location assigned to the selected subscription or simply visualize its position on the map.

Client specific tabs: by selecting a client in the relative list, the tabs below the list will get filled with the information available for that client. The user may want to simply visualize them and optionally modify them.

- Personal data: personal information related to the client (name, surname, category of client and so on);
- Sea Pass: this is the name given to the RFID tag within the context of MIIB's project. So this tab gives access to the functionalities of tag assignation and substitution.

- Virtual account: through the controls present in this tab the box office employee is able to recharge client's virtual account, modify the associated PIN and visualize all the transaction made on the account.
- Interests: the interest of the client can be visualized and modified in this tab.
- Services and activities: additional services and activities bought or to buy are manageable in this tab.

Check out: a single client may want to leave the subscription by giving back his/her personal tag.

Guest adding: a certain number of guests is allowed on a sold location. So through the relative button, user can access to the relative Guest Adding view.

Reprint of receipt: the receipt printed at the moment of the subscription can be reprinted.

Subscription termination: this procedure is necessary when a subscription is over and clients bring back their tags. After procedure completion, the subscription is considered closed and no more visible.

Some other major functionalities are reachable through the Administration and Tool form. These functionalities cover the need for printing detailed data on what have been sold during the day, in order to send them to the accounting office along with the cashed money. A **turnstile management** section is also present, so that the box office employee is able in any time to verify turnstiles functioning and eventually perform a manual piloting of the blocking engine.

Code structure

Box Office module bases its functionalities on an set of Controllers that offer to it different services to retrieve, process, store data, to execute complex algorithms or to interface to the RFID reader. All these services are offered by different business layer components also named Controllers. We decomposed all these services in many controllers considering Box Office High level functionalities. All these controllers are contained in a package named CustomComponent.Ctrl.BoxOffice. CustomComponent.Ctrl contains also other packages that contain controllers of other two subsystems(IT Kiosk, Remote interface). Therefore controllers we designed for this module are:

- AdministrationCtrl, contains methods to manage the turnstiles manually and print daily cash flows
- ClientManagementCtrl, contains methods to manage the Client management section by what searching Customer's profile and subscriptions info
- DailyAccessCtrl, contains methods to manage the Daily Access procedure that allows a customer to subscribe a Daily access.
- LoginCtrl, contains methods to execute the Employee login procedure

- NewSubscriptionCtrl, contains methods to manage the Periodic Access procedure that allows a customer to subscribe a periodic(week, month, trimester etc.) subscription.

Due to the big number of functionalities offered, and the Presentation layer structure that allows employee to execute many contemporary subscription procedures, a particular Caching strategy had to be chosen to reduce the redundant data loading operations and increment application responsiveness and performances. The most important functionalities, new and daily subscription, and client management procedures require In fact a large amount of static data. Instead of being loaded everytime a new procedure is started, these data could be loaded at application start inside a cache to be updated rarely, only when modifications (subscription completions, info modified) occur. We decided therefore to create an object in charge of managing this data cache, called CacheCtrl. So, the controllers listed above will load static data from the cache and not from the database. The MDI model chosen to manage the GUI components, that allows Employee to execute contemporary tasks(client management, new or daily subscription procedure) pushed us in defining a controllers management strategy. Every procedure should In fact use only one controller, and only of the type related to manage that procedure. The possibility of executing many similar procedures, and the step structure of new and daily subscription procedure, requested to create pools of controllers generated by a controllers factory. So, every time a procedure is started:

- A new controller is created by the factory and put into a same type controller pool.
- In this way, windows of every procedure can catch, when needed , its controller from the pool. These features give us many advantages:
 - Every controller can be easily caught from any element of the same procedure
 - we minimized the number of controllers for a procedure(only 1 for each type)

Presentation Layer The Box Office is the module with the most structured presentation layer. This is due to the kind of tasks intended for this subsystem. The interface is based on a structure called multiple-document interface (MDI) and it's a way of setting up a user interface for applications that enable the user to work with more than one document simultaneously. The peculiar User Interface Components in an MDI application are of three kinds:

- frame window: it has to be considered as the main window of the application. It's characterized by a sizing border, a title bar, a window menu, a minimize button, and a maximize button.
- MDI client window: it's a child of the frame window and it belongs to the pre-registered window class MDICLIENT. The purpose of this window is to serve as a background for child windows and to provide methods for creating and manipulating child windows.

- a number of child windows: they are windows whose visualization is limited inside the boundaries of the frame window. They all see the client window as their parent.

In the Box Office module the ParentForm is the main window of the application and all child windows are visualized inside its boundaries.

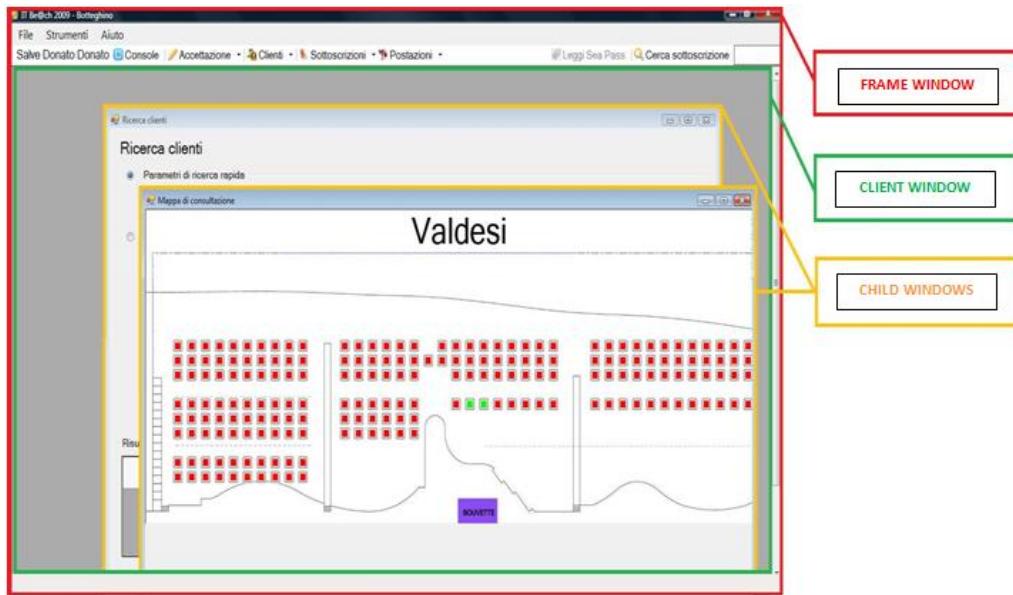


Figure 6.7. MDI structure

As explained in paragraph 6.4 (Presentation Layer), it's recommended to extrapolate interaction logic from User Interface (UI). Some developers, unaware of the benefits deriving from responsibilities separation between User Interface Components and User Process Components, put all the logic within the user interface code itself. By doing this the resulting code can be complex and very difficult to reuse, maintain or extend. In addition, it could be quite hard to bring the application on another platform since application control logic and state cannot be reused. Furthermore, box office's procedures requires a state to be maintained. Storing state within forms implies that a reference to that form must be retrieved every time a data in the state is needed. This can be difficult to implement and tends to result in inelegant code, which again affects the extensibility and reusability of a user interface. A good example of low flexibility regards multi-step procedures in which each form have to be viewed sequentially. If in some point during development or, in the worst case after application completion, the developer wants to insert a new step in the sequence, he/she will have to recode or modify both the previous and subsequent forms to incorporate the new form. It is thus clear that the best choice in developing the box office module was to consider workflow, navigation, and interaction with business components as separate concerns from how data is acquired and presented to the user.

Among the Application Blocks developed by Microsoft (<http://msdn.microsoft.com/en-us/library/aa480453.aspx>) we decided to make use of the second version of the User Interface Process Application Block (UIPAB) which provides an extensible framework to simplify the process of separating business logic code from the user interface. Since we decided to implement New Subscription procedure and New Daily Access procedure as wizard workflows, the need for an optimal state persistency and workflow control arose. UIPAB resulted to be a suitable support since it addresses easily these two challenges. In general, UIPAB supports:

- Navigation and workflow control - This should not be embedded in the user interface, but often is because the decision about which view to display next is based on business logic. This results in inelegant and unmanageable code.
- Navigation and workflow changes - Reformatting the layout of an application (changing the sequence of forms or adding new forms) is very difficult with traditional user interface techniques.
- State management - Passing state and maintaining consistency of state between views is difficult and is different for Windows-based applications and Web applications.
- Saving a snapshot of current interaction - You may want to capture a snapshot of an interaction and recreate it elsewhere, across time, machine, or logon boundaries.

An analysis of the UIP Application Block reveals that the model-view-controller (MVC) pattern is implemented. This is a fundamental design pattern for the separation of user interface logic from business logic. The pattern separates the modeling of the application domain, the presentation of the application, and the actions based on user input into three distinct objects (as shown in figure 6.7):

- Model -this object is aware of all the data to be displayed and is responsible for managing the data and the actions of the application. It can be thought of as the processing part of an input-process-output system.
- View - This object manages the information displayed to users. Multiple views can be used to display the same information in different ways. It can be thought of as the output part of an input-process-output system.
- Controller - This object allows the user to interact with the application. It takes input from the user and passes instructions to the model. In the input-process-output system, this is the input part.

The UIP Application Block contains several classes and interfaces that combine to provide the desired infrastructure. The figure 6.8 illustrates the design of the UIP Application Block, showing the main components of the block, and how they interact with components provided by your organization.

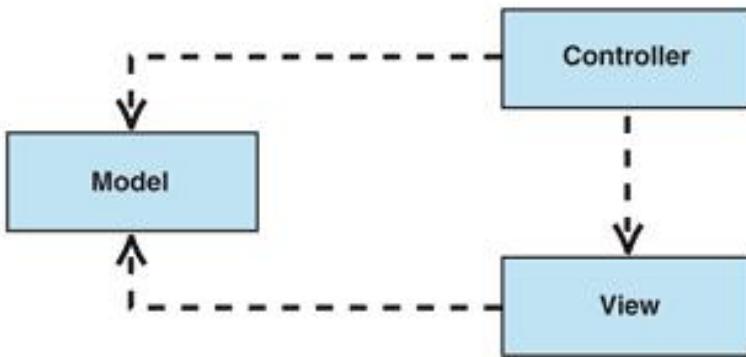


Figure 6.8. The Model View Controller Pattern

UIP Application Block Elements:

Your application: this is the Box Office module containing the calls to task start methods;

Configuration file: contains application configuration information;

Your controllers: these are the objects controlling the navigation between views and act as a façade between the Presentation Layer and the Business Layer. They inherit from Controller Base;

Controller Base: abstract class useful to create our controllers;

Services, Business Components: Services and business logic of Box Office module;

UIP Manager: provides entry points to the UIPAB for starting or loading a task with a variety of navigators;

Navigator: is responsible of coordinating the navigation among views and determines which view to activate at each transition, asking the View Manager to activate it;

UIP Configuration: Retrieves, verifies and stores the information contained in the application configuration file (App.config in our Module);

Your Views: User Interface Components with which the user will interact;

Views: View classes useful to create our views;

View Manager: Class responsible for Views creation and activation upon navigator's request;

State: This class is responsible for maintaining the current view in the task and user process state; in our Module, the state is used to keep the information provided by the user during while arranging a new subscription or new daily access; State Persistence Provider and State Persistence Store: these elements are useful when the application should be able to save a snapshot of the current task to let it be reloaded in a later moment. Since this feature is not required in our context is not useful to spend time explaining their characteristics.

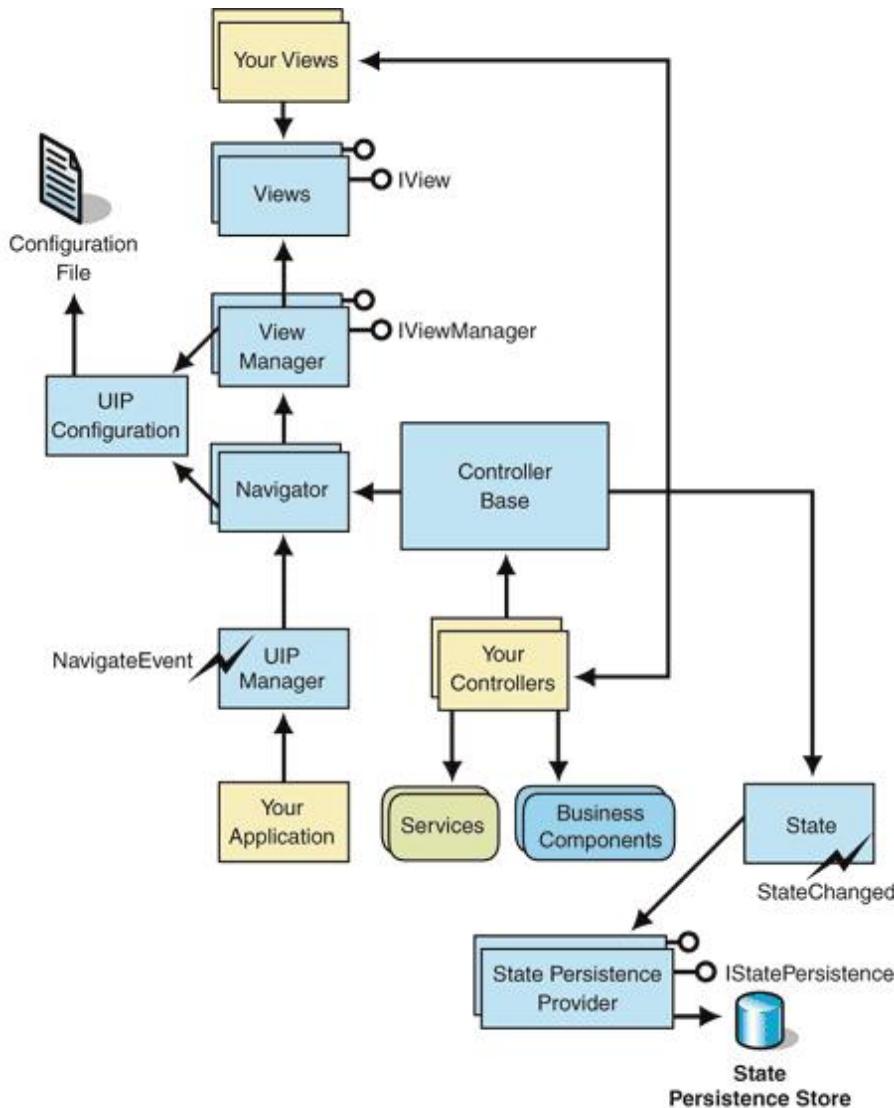


Figure 6.9. The UIPAB infrastructure

One of the earliest challenges we had to face was the integration of UIP Application Block with the MDI infrastructure. One of the fundamental steps in MDI schema deployment is to explicitly set at runtime the correct parent form for the child window before it's displaying. Since Views creation and visualization is managed inside the UIP Application Block, there was no way of setting this property. The result was a wrong visualization having the created view to be displayed outside the scope of frame window. The only way to get rid of this shortcoming was to manipulate the UIPAB customizing it to our needs. After finding the correct point in which the view to display was created, we still had to let a reference to the parent form reach that portion of code. With a chain of passages this

has been made possible, and the only visible modification at the application end side is an overload of the StartNavigationTask method of the UIPManager class accepting an object TaskArgumentsHolder as argument. Before calling this method, a reference to the frame window is put in the TaskArgumentsHolder object. Let's now continue on the explanation of what happens when the StartNavigationTask method get called:

- After a task is started, creation of views, navigation, and state management are performed through UIP.
- The UIP manager creates the appropriate navigator (we implemented four different navigators for the more complex tasks provided for this module) and passes the necessary information to the navigator.
- The navigator calls the UIPConfigSettings class to retrieve configuration information parsed from the configuration file.
- The navigator creates the appropriate view manager, based on the type of views being used (as specified in the configuration file). In our interface, the views inherits from the WindowsFormView class.
- The navigator creates the task determining which view to activate first.
- The navigator issues a NavigateEvent event through the UIP manager. This provides the current and next view and can be used by applications external to the UIP Application Block.
- The navigator calls the ActivateView method on the appropriate view manager and passes the view name to the view manager. The ActivateView method is responsible for creating the controller and displaying the view.
- After the view displays, the user is in control and UIP does nothing further until the user performs an action on the user interface (for example, clicking a button).
- The user performs an action, and the event causes a method to be called on the view, which in turn calls a method on the controller. Depending on the user action, the controller does one of the following:
 - Perform or delegate a function related to business logic (such as saving information to the database) that causes the view to refresh and change state. If the function causes a modification to the current state, the State class will raise the StateChanged event, to notify any interested listeners of changes that have occurred to the state. We don't use this feature in our module.
 - Call the Navigate method on the navigator to transition to the next view or user control, and update the State object.

The first step towards the implementation of this presentation layer architecture is the arrangement of the Configuration File. The UIP Application Block provides an XML Schema

Definition (XSD) file called UIPConfigSchema.xsd, against which our App.config file has been validated. This schema details which elements and attributes are required and which are optional, how many times each may occur, and in what order they should occur. Since the App.config file is needed also for other purposes, the code useful for UIP configuration constitutes only a part of it and it's limited inside a section called `<uipConfiguration>` The structure of the `<uipConfiguration>` section is shown in the following code. The sections in *italics* are optional.

```
<uipConfiguration>
  <objectTypes>
    . . .
  </objectTypes>
  <views>
    . . .
  </views>
  <sharedTransitions>
    . . .
  </sharedTransitions>
  <navigationGraph>
    . . .
  </navigationGraph>
  <uipWizard>
    . . .
  </uipWizard>
  <userControls>
    . . .
  </userControls>
</uipConfiguration>
```

The sections `sharedTransitions`, `uipWizard` and `userControls` are not used in our module so they won't be explained. Just a note is worth making about the `uipWizard`: this section is needed to take advantage of a feature of UIPAB concerning wizard-like tasks; though New Subscription and New Daily Access procedures are organized in a wizard style, the support provided by UIPAB was not suitable in our case. First of all, the visual presentation was like the wizard used by Windows Installer, so quite small to host all the information needed in every step of the procedure. The other feature is the automatic arrangement of views in a Next-Previous relation (easy to obtain in a customized navigator). Furthermore, the wizard defined by the `< uipWizard >` element supports linear navigation, and the wizard defined by the `< navigationGraph >` element supports branching. So we preferred to avoid UIPAB's wizard and kept on developing our own wizard. Let's now take a look on the other sections, the ones we actually used in our Module:

- **objectType**: Defines the classes to be used for managing and controlling the user interface process in an application.
- **views**: Defines the details of the views to be used in an application.

- **navigationGraph:** Defines the details of the graph navigator. Required by each task that uses a graph navigator. Used to define wizards that support branching.

The first section to define is thus the objectTypes one:

```
<objectTypes>
  <iViewManager . . . />
  <state . . . />
  <controller . . . />
  <layoutManager . . . />
  <statePersistenceProvider . . . />
</objectTypes>
```

In our module we needed to set only the first three parameters of this section so the use for layoutManager and statePersistenceProvider will not be explored. iViewManager: specifies the view manager responsible for activating and deactivating views within the application. state: controls the state of your application. Can be extended to include application-specific state attributes. controller: controls the flow of the views and responds to requests from views.

Each of these sections requires specific attribute settings to link it to the relevant class in the UIP Application Block:

- name: the name used to refer to this object within the configuration file and block code.
- type: the type information for this object, which identifies the assembly where the object is defined and its class name.

The following code reports the < objectTypes > section for the Box Office Module:

```
<objectTypes>
  <iViewManager name="WindowsFormViewManager"
    type="Microsoft.ApplicationBlocks.UIProcess.WindowsFormViewManager,
    Microsoft.ApplicationBlocks.UIProcess,
    Version=1.0.1.0,Culture=neutral,PublicKeyToken=null" />
  <state name="State" type="Microsoft.ApplicationBlocks.UIProcess.State,
    Microsoft.ApplicationBlocks.UIProcess, Version=1.0.1.0,Culture=neutral,
    PublicKeyToken=null" default="true" />

  <controller name="UIController" type="BoxOfficeUIController.UIController,
    BoxOfficeUIController, Version=1.0.0.0,Culture=neutral,PublicKeyToken=null" />

  <statePersistenceProvider name="MemoryPersistState"
    type="Microsoft.ApplicationBlocks.UIProcess.MemoryStatePersistence,
    Microsoft.ApplicationBlocks.UIProcess, Version=1.0.1.0,Culture=neutral,
    PublicKeyToken=null" default="true" />
```

```
</objectTypes>
```

As mentioned before, statePersistenceProvider is not used in our Module since we don't want to enable task resuming after it is closed. Nevertheless, the section is mandatory, so we specified the MemoryPersistState provider, managing state storage on local disk.

The `< views >` section defines the details of the views used in the Module that requires UIPAB's support. Each of these views correspond to an element of this section and requires three attributes: the name, type, and controller. name: the name of the view as referred to in the navigator. type: the class name of the class that defines this view. controller: the name of the controller as defined in the `<objectTypes>` section to be used for this view.

Here's an example of the views section elements defined in our Module:

```
<view
  name="Step1"
  type="BoxOffice.Step1, BoxOffice, Version=1.0.0.0, Culture=neutral,
  PublicKeyToken=null"
  controller="UIController"
/>
```

The `<navigationGraph>` section defines the details of the graph navigator. It also declares the objects to be used for navigation management. The important attributes in this section are: startView: the name of the first view to be shown within the process. Note that this is not necessarily the first view that the user sees, but the first view referenced in the graph navigator. iViewManager: the name of the view manager as defined in the `< objectTypes >` section. name: the name of the graph navigator. state: the name of the state type as defined in the `< objectTypes >` section. statePersist: the name of the state persistence mechanism as defined in the `< objectTypes >` section.

The `< navigationGraph >` section also contains a number of `< node >` elements, each of which define a node within the graph navigator. The view attribute of the node identifies the view as defined in the views section, and a certain number of `<navigateTo>` elements identifies the navigational routes for this view. The `< navigateTo >` elements have two attributes: navigateValue: the value that is assigned to an action in the code to identify the route to be taken. view: the view that should be navigated to when this navigateValue is used.

To be more clear of navigator features we propose here one of the four navigators created for the Box Office Module, more specifically the one intended for the New Subscription procedure:

```
<navigationGraph iViewManager="WindowsFormViewManager"
  name="NewSubscriptionNav" runInWizardMode="false"
  startView="Step1" statePersist="MemoryPersistState" state="State">
  <node view="Step1">
    <navigateTo navigateValue="next" view="MapView" />
```

```
</node>
<node view="MapView">
    <navigateTo navigateValue="previous" view="Step1" />
    <navigateTo navigateValue="next" view="Step3" />
</node>
<node view="Step2">
    <navigateTo navigateValue="toMapView" view="MapView" />
    <navigateTo navigateValue="next" view="Step3" />
</node>
<node view="Step3">
    <navigateTo navigateValue="toMapView" view="MapView" />
    <navigateTo navigateValue="next" view="Step5" />
</node>
<node view="Step4">
    <navigateTo navigateValue="previous" view="Step3" />
    <navigateTo navigateValue="next" view="Step5" />
    <navigateTo navigateValue="toServices" view="ServicesView" />
</node>
<node view="ServicesView">
    <navigateTo navigateValue="back" view="Step4" />
</node>
<node view="Step5">
    <navigateTo navigateValue="previous" view="Step3" />
</node>
</navigationGraph>
```

This navigator is a perfect example to show how easy it would be to add a view in the sequence of views used to execute the task. As you may notice with a attentive look, Step4 and ServicesView are actually unreachable since the navigation path proceeds from Step3 to Step5 and then from Step5 to Step3 again. This fact has an explanation: during system requirements elicitation and analysis, MIIB showed us the intention to have support for additional services to sell during New Subscription procedure. Then the support was provided but no service was ready to be sold during development. The ServicesView has been thus temporarily excluded from the path. Similar consideration can be made for Step4 but here the point is: it's very easy to add or remove views from the navigation path and this is one of the most powerful features of the UIP Application Block.

Concurrency Another key development hot spot was the arrangement of a robust concurrency model. In the first scenario we designed, we made the hypothesis of two concurrent Box Office Modules operating simultaneously. A lock mechanism on location has been implemented. It is based on the insertion of a record in the Subscription table for the location selected at the second step (Map View). This record has a flag denoting that the procedure is still not completed but, for the other Box Office Modules checking

location availability for the same period, the location results busy. If the subscription gets confirmed, the flag on the relative temporary record is flipped and thus made permanent. Otherwise it is removed from the table, which results in freeing the relative location. According to the explained mechanism some additional considerations have to be made:

- If two Box Office Modules arrives simultaneously on the Map View after selecting two overlapping periods for a new subscription, and if they both try to make a reservation for the same location, the second who arrives to select it will get a notification that the location was already selected by another Box Office.
- Since the features to be set at Step 1 of a New Subscription change considerably the resulting location availability, every time user wants to return to Step 1 from the Map View after a location had been already reserved, the application will notify the user that by executing the transition the reservation will be cleared.

One may argue that this model introduces a problem whenever a user is in the middle of a New Subscription or Daily Access procedure and loses connectivity to database or experiences an application/system crash, this will leave the database in an inconsistent state with a location incorrectly busy. This is why we inserted a column in the Subscription table to host the computer name that performed the insertion. Every time the Box Office Module restarts, it will brutally delete from the database the Subscription record with the host name corresponding to the machine where it's running and the inCreation flag set to true. This will bring the database in a consistent state again.

Hardware Deployment

The Box Office module working requirements, determined partially at Design phase, and its functionalities implemented and mostly illustrated in this chapter, conducted us in identifying and choosing the devices useful to allow this subsystem to run.

The identification of most of these components, used anyway in every other subsystem, especially the RFID related ones, was not simple since we performed a careful research in finding devices with the highest quality/price ratio focusing more on minimizing project's costs. We are proud of communicating that we finally found really cheap RFID devices from out of Italy and out of Europe suppliers. Prices resulted to be halved price of those ones proposed from many Italian suppliers.

In the table below we list all the components chosen for Box Office module running, reporting quantity, device class, model name, the related supplier and a brief description.

6.5.2 Remote interface

Introduction

The Remote Interface module is the interface module to fulfill the persistent data used in the whole system. We already analyzed the functionalities and the design of this module. As we already said, this application is developed as a 3 layered rich client application. The data it has to deal with is:

Q.ty	Device category	Model name	Supplier	Description
1 x	Laptop	Acer Aspire 5735Z	Euronics (Italy)	Acer laptop characterized by the T3400 Intel Pentium dual core processor(2,16 GHz/1M); 15.6" LCD screen; 3 GB DDR2 RAM; 250 GB HD; DVD reader; shared memory graphic card; 1Gb Ethernet Network interface; Windows Vista Home Premium OS.
1 x	RFID reader	Arygon, ISO 14443-NFC compliant, USB Desktop reader	IDTronic (Germany)	USB RFID desktop reader compliant with NFC and ISO 14443A standards. It is used to read MIFARE 1K RFID tags we chose.
1 x	Receipt printer	Custom KUBE ETH, 80mm	Custom (Italy)	80 mm Receipt printer used to print no-fiscal receipts and coupons.

Table 6.1. BoxOffice hardware components

- Type of locations
- Tags
- Beaches: for each beach, the following informations are handles:
 - Categories of clients
 - Opinions
 - Types of subscription
 - Activities
 - Services
 - News
 - Employees
 - Discounts
 - Map of locations

Interaction organization

As for the Box Office module, the interface is based on a structure called multiple-document interface (MDI) and it's a way of setting up a user interface for applications

that enable the user to work with more than one document simultaneously. The peculiar User Interface Components in an MDI application are of three kinds:

- frame window: it has to be considered as the main window of the application. It's characterized by a sizing border, a title bar, a window menu, a minimize button, and a maximize button.
- MDI client window: it's a child of the frame window and it belongs to the pre-registered window class MDICLIENT. The purpose of this window is to serve as a background for child windows and to provide methods for creating and manipulating child windows.
- a number of child windows: they are windows whose visualization is limited inside the boundaries of the frame window. They all see the client window as their parent.

In the Remote Inteface module the ParentForm is the main window of the application and all child windows are visualized inside its boundaries. In order for the employee to access the module, he/she has to authenticate: therefore, the first thing that has to be done is to fill the login form with a valid username and password. After the user is logged in, the first windows shows the functionalities not connected to any beach in general, as we can see in the figure below:



Figure 6.10. First screen of the Remote Interface after the login

The employee can therefore access to:

- **Report:** with this section it's possible to visualize the report and eventually print them.

- **Beach:** with this functionality it's possible to manage the CRUD operations for the Beaches, and to handle their maps.
- **Types of locations:** with this functionality it's possible the CRUD operations for the types of locations, and to associate an icon for each type of location (e.g. a sunbed, a palm, a bungalow...)
- **Tag:** with this functionality it's possible to manage RFID tags that will have to work within the system.

Otherwise, it's possible to access to the area dedicated to the functionalities related to the information that are related to a certain beach after selecting a beach from the combobox, as we can see in the figure below:

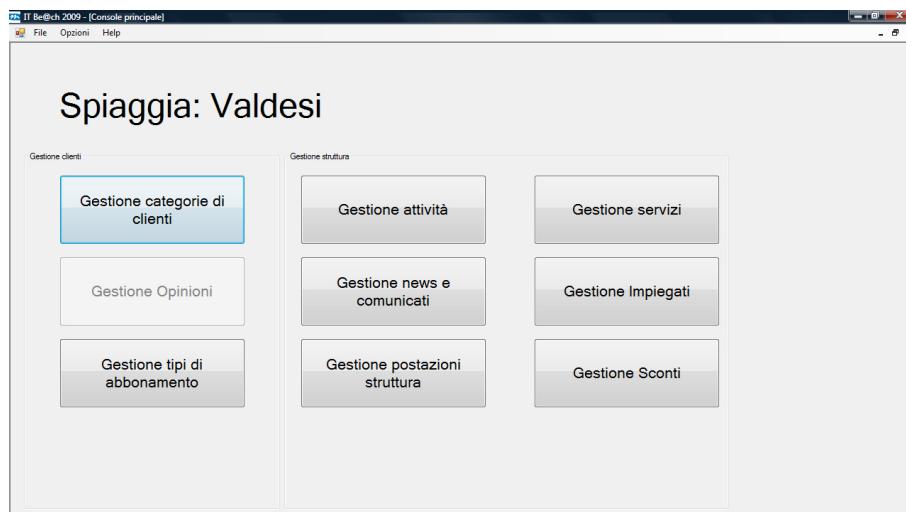


Figure 6.11. The windows of the Remote Interface for the management of data related to a certain beach.

By clicking any button in this section, a page for CRUD operations will be showed. An example is shown in the figure below:

As we can see from the picture, the organization of the data management is handled this way:

- A listbox in the left side is indicating the list of the items actually inserted in the database
- The “insert” button can be used to insert a new item.
- The “modify” and “delete” buttons can be used to modify or delete (if possible) an existing item
- All the information about the item can be seen in the right side of the window

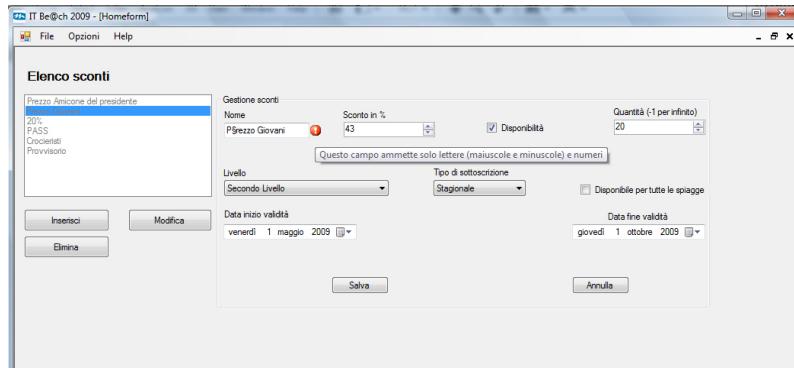


Figure 6.12. Discounts management: an example of CRUD page

When inserting or modifying an item, a validation control is made in every field. If any of the field is not valid (for example because a number is present in a field that accepts only chars), an error message is shown and it's not possible to carry out the requested operation (we can see an example in the figure above: trying to insert a character which is not part of the english alphabet in the "name" field raises a validation error).

The management of locations in a map is graphical and therefore not typical as the one described above. A detailed description will be given in the "development hot spots" paragraph of this module.

Code structure

Since this module was designed to follow the 3-layered architecture, it was organized in the following way:

- A visual studio project contains all the windows forms composing the module
- A class library project called CustomComponents contains the business layer: all the algorithms and the libraries to connect to the hardware devices are contained in this library. This class library project includes also all the libraries related to the data layer.

One controller in the CustomComponents library was created for each functionality:

- ActivitiesCtrl: used to manage operations related to the activities;
- BeachesCtrl: used to manage operations related to the beaches;
- CategoriesOfClientCtrl: used to manage operations related to the categories of clients;
- CustomEntitiesCtrl: this class contains some entities that are created by purpose and not directly from the data layer.

- EmployeesCtrl: used to manage operations related to the employees;
- LocationsCtrl: used to manage operations related to the locations;
- NewsCtrl: used to manage operations related to the news;
- ServicesCtrl: used to manage operations related to the services;
- SubscriptionDiscountCtrl: used to manage operations related to the subscription discounts;
- TAGCtrl: used to manage operations related to the tags;
- TypesOfLocationCtrl: used to manage operations related to the types of locations;
- TypesOfSubscriptionCtrl: used to manage operations related to the types of subscriptions;

Development Hot Spots

Locations map management As we already mentioned, the maps of locations for beaches are managed in a graphical way. For each beach, a map is filled with locations that can be of different types and can be represented by different images. The types of locations can be of two different types:

- Ornamental: these types of locations are just ornamental (it can be a palm tree, a wall..) and therefore cannot be selected as a location during the subscription procedure in the box office.
- Usable: these types of locations can be used during the subscription procedure. A different price or discount can be applied depending on different types of locations (e.g. first level locations, second level locations).

The manage map locations section is organized with 4 functionalities:

- Insert a new location in the map
- Modify the data about a location in the map
- Modify the position of a location in the map
- Delete a location from the map

Before accessing this section, a map for the beach has to be inserted in the "beach management" functionality.

In order to create maps that are precise as much as possible, we created an algorithm that creates a virtual grid to alignate in an automatic way the locations, both when inserted and when modified the position. The algorithm works as explained here:

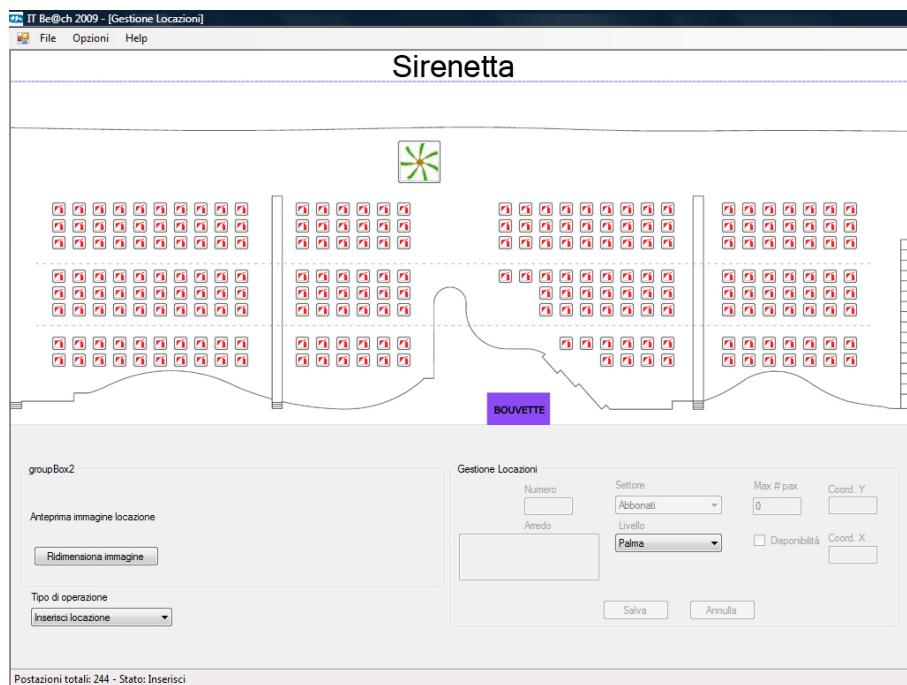


Figure 6.13. Insertion of a new location

- At first an empty map is showed
- When the "insert" functionality is selected, the icon related to the type of location is showed when moving inside the map; in this way the user knows exactly where the location will be inserted. (figure 6.7)
 - The icon representing the type of location, is resizable: a functionality dedicated to the resize is available in a new window by clicking to a button. Obviously in the database information about the size of a icon relating the location: in this way, when designing a map in the Box Office, there are all the information about how to draw the correct dimension of the locations.
- After the user select the position desired and click, the system:
 - Calculate the right position based on the virtual grid associated to the map. The position is calculated from the bottom-left corner of the icon. (figure 6.8)
 - After calculating the right position, the algorithm checks if the icon would overlap another location's icon already saved in the database. If this happens, an error message is shown and the insertion is not completed. Otherwise, the location can be inserted in that position.
- The location is still not saved in the database: before inserting it, the user has to complete some information about the location (e.g. number, furniture..).

- After inserting these values, if there are no validation errors, on user's saving the location is inserted into the database with the information about it (both graphical information such as coordinates and management information).

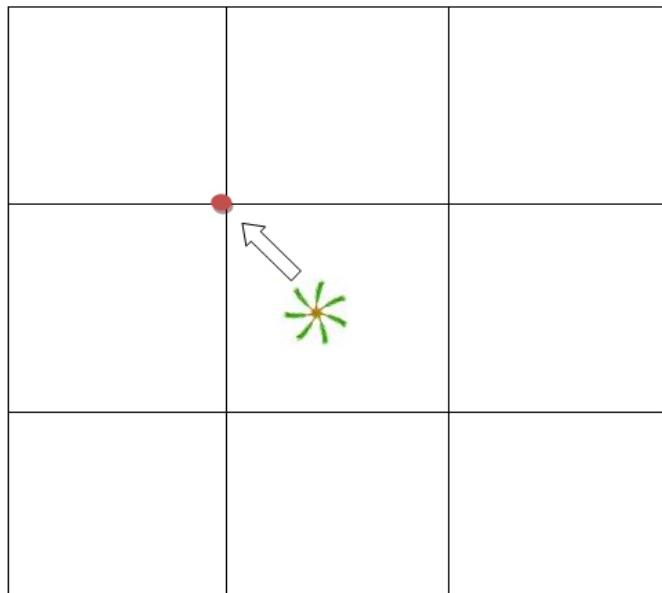


Figure 6.14. Schema of the algorythm to calculate the virtual grid

Deployment and Hardware

The Remote interface module working requirements, determined partially at Design phase, and its functionalities implemented and mostly illustrated in this chapter, conducted us in identifying and choosing the devices useful to allow this subsystem to run.

In the table below we list all the components chosen for Remote interface module running, reporting quantity, device class, model name and the related supplier and a brief component description.

6.5.3 Bar payment module

The Bar payment module is a subsystem used to realize the RFID tag payment paradigm. As already mentioned during its design paragraph, it should collaborate with an off the shelf Bar management software, RIGEL. This allows Bar employees to manage orders, lounge tables and close orders after paid. The only way of payment it provides is the cash or coins one. The Bar payment module goal is therefore enlarging its functionalities allowing RIGEL to execute RFID tag payments. Communicating only with the RIGEL system, offering it a new functionality(a service), it was developed as a Service Application.

Q.ty	Device category	Model name	Supplier	Description
1 x	Laptop	Acer Aspire 5735Z	Euronics (Italy)	Acer laptop characterized by the T3400 Intel Pentium dual core processor(2,16 GHz/1M); 15.6" LCD screen; 3 GB DDR2 RAM; 250 GB HD; DVD reader; shared memory graphic card; 1Gb Ethernet Network interface; Windows Vista Home Premium OS.
1 x	RFID reader	Arygon, ISO 14443-NFC compliant, USB Desktop reader	IDTronic (Germany)	USB RFID desktop reader compliant with NFC and ISO 14443A standards. It is used to read MIFARE 1K RFID tags we chose.

Table 6.2. RemoteInterface hardware components

Interaction organization

Considerations The Bar payment module interacts with the RIGEL system to execute RFID tag payments when an order has to be closed. A perfect synchronization should be persecuted. When the Bar employee confirms the order closing and the related payment, using RIGEL functionalities, our Bar module would intervene immediately activating the RFID reader to read customer's tag and retrieve all information related to his/her credit account. After the appropriate checks, transaction can be completed. Let's decompose the interaction problem in several steps to better explain in the Steps Interaction Analysis paragraph how we intervened. After ordered and consumed:

1. Customer asks to close his/her order, pushing the employee to close it through RIGEL functionalities. A total amount is calculated and the employee selects the RFID tag method of payment.
2. Employee invites customer to insert his/her PIN, and after it confirms the procedure.
3. Only after the confirmation, Our bar payment module intervenes gathering(receiving) the PIN, total amount and RIGEL transaction code information, and activates the RFID reader to read customer's tag.
4. Customer passes RFID tag over reader working field allowing bar module to retrieve the data related to his/her credit account.
5. After the appropriate checks, it confirms or refuses the transaction to RIGEL.
6. RIGEL visualizes the transaction outcome.

Steps Interaction Analysis Before arriving to the definition of the interaction strategy, we performed a strict collaboration with RIGEL developers to define the most appropriate way of synchronizing and exchanging information. To better understand how face problems practically a brief study of the original RIGEL functionalities was required, focusing more on the ones related to the order closing and payment. We therefore installed a demo version of it. Then, we started to analyze every single step necessary to complete the whole interaction, in order to discover RIGEL major issues and face them efficiently. Obviously the interaction could be obtained only by introducing modifyings to the original RIGEL version.

Step1 RIGEL allows employee to close and pay an order, showing him/her a popup containing two different buttons by what employee indicates the following payment methods:

- Cash or coins
- Coupons

Clearly a new button had to be added to allow RFID tag payments. The selection of this button was necessary to pass to step 2.

Step2 Again, a new modifying had to be done to let customer inserting the RFID tag PIN. The selection of the RFID payment button at step 1 should bring RIGEL to the PIN insertion state. Using a Pin pad device, customer should insert his/her PIN and confirm, passing to Step 3.

Step3 The password insertion confirmation allows RIGEL to exit PIN insertion state to finally communicate our Bar payment module the data needed. A communication method and synchronization strategy had to be chosen in order to send :

- Customers' PIN, useful to execute payments safely. It avoids that eventual stealers enjoy RFID tag money.
- RIGEL transaction code, RIGEL calculates a temporary transaction code to allow our Bar payment module to register it in our database. It allows management to execute successive databases(RIGEL one and ours) crossing checks when desired.
- Total amount, bar module should know how exactly decrease customer's credit account. Received all these information, the Bar payment module activates the RFID reader to read customer's RFID tag, passing to step 4.

Step4 Customer passes RFID tag over reader working field allowing bar payment module to get and use RFID UID for retrieving customer's credit account information.

Step5 Bar payment module checks the password validity, the present credit amount, confirming the payment only if he owns enough credit to cover the total amount required. It prepares the transaction outcome to confirm or refuse the transaction. Again, a new information exchange task is needed to let RIGEL visualize the transaction result at step 6.

Step6 RIGEL receives the transaction outcome and visualizes it to the employee. A popup message is required.

Development HotSpots

To perform a fluid and efficient interaction between the two modules we made different implementation choices concerning bar payment module overall structure, information exchange methods, and synchronization.

Overall structure We decided to develop this module as a Windows Service Application since its only task was providing RIGEL a service, a new method of payment. Service Application could us guarantee high performances and availability since the module can be started automatically at the OS bootstrap. This lets bar payment module to fit perfectly RIGEL infrastructure. Information exchange method The execution of Steps 3 and 5 imply the communication of useful data between the two subsystems. The most common ways of passing data through applications are realized using sockets and files. The socket communication is often used for a remote communication, situations in what different systems are deployed in different tiers/machines. Files communication, instead, can be used from processes that run on the same machine. Considering that RIGEL and our bar payment module are deployed in the same tier(Bar PC), we decided to use a File based communication. The data sender for steps 3 and 5 should thus format the dataflow on a .txt file allowing the receiver to correctly retrieve every information it contains. For this reason, we agreed with RIGEL developers on:

- two file formats, one for each of the two steps
- a shared/communication working directory, containing the files to exchange

In the figure below we show the data format schema.

Fig.: exchange files format. On the left the step3,request of payment, and on the right, step 5, transaction outcome.

Synchronization The activation of the RFID reader at step 3 requires the choice of a perfect synchronization strategy between the two modules. The bar module should In fact activate the RFID reader only when RIGEL sends a particular signal. So, considering that in addition to a signal, RIGEL should send a request of payment(step 3), the real signal could become in this way the .txt file containing the request information. How can bar module notice the presence of file? Since it will not use sockets, it has to monitor continuously the shared folder that hosts the temporary request file. When the file is

created, it stops polling, reads it, memorize the information read in local variables, and cancel the file, to leave the folder free for next request of payments. Now it activates the RFID reader waiting for the customer. Particular attention was posed on this waiting status. If the customer doesn't pass the RFID tag over the related reader, this task never ends, causing the locking of the application. For these reason, we decided to use a Timer by what defining a maximum listening time interval. Even if the customer loses time or renounces to pass the RFID tag, after a certain time interval, the application ends listening, returning the transaction outcome(an error occurred in this case). Bar payment module is again available for any other payment. Assuming the customer passes his/her tag, Bar payment module continues its task retrieving customer's account data, and after appropriate checks on credit amount and PIN validity, it answers RIGEL creating a new .txt file in the shared/communication folder. With the similar synchronization mechanism, RIGEL notices the file receiving and visualizes the transaction outcome.

Code Structure

Bar payment module was developed as a Windows Service application that starts automatically at the OS bootstrap. It is characterized by a class named BarService that contains the common Service applications methods, OnStart() and ONStop(), and a method called watcherFileCreated(...) that is registered to the file created event handler of the FileSystemWatcher object used to manage the folder monitoring. This method is invoked when a new file creation event occurred, and it owns all the logic core. We divided the logic core in the following steps:

- Request File and tag reading, tag reading is executed using our private RFID interface library, developed for the desktop readers purchased.
- Data retrieval and transaction checks, every necessary check is executed before concluding the payment(PIN validity, tag assigned, sufficient credit, commercial transaction registered).
- Answer File creation, the answer file is created and formatted to returns the payment outcome.

Hardware Deployment

The Bar payment module working requirements, determined partially at Design phase, and its functionalities implemented and mostly illustrated in this chapter, conducted us in identifying and choosing the devices useful to allow this subsystem to run.

We remember it should interact with RIGEL software system, chosen to manage the Bar orders and payments, in order to provide it the new RFID tag payment method. Since it's necessary to deploy them on the same machine, the identification of the components have to take into account either RIGEL and Bar payment module requirements.

In the table below we list all the components chosen for RIGEL and Bar payment modules running, reporting quantity, device class, model name, the related supplier and a brief component description.

Q.ty	Device category	Model name	Supplier	Description
1 x	Desktop PC	HP KV364ET	Informatica Commerciale (Italy)	HP Desktop PC characterized by the E7400 Intel Pentium dual core processor(2,66 GHz/1M); 3 GB DDR2 RAM; 250 GB HD; DVD reader; shared memory graphic card; 1Gb Ethernet Network interface; 17" LCD Touch screen; Windows XP Home Edition OS.
1 x	RFID reader	Arygon, ISO 14443-NFC compliant, USB Desktop reader	IDTronic (Germany)	USB RFID desktop reader compliant with NFC and ISO 14443A standards. It is used to read MIFARE 1K RFID tags we chose.
2 x	Receipt printer	Custom KUBE ETH, 80mm thermal printer	Custom (Italy)	80 mm Receipt printer used to print RIGEL orders for the Bar employee.
1 x	Uninterruptible Power System(UPS)	APC, Schneider electric	Informatica Commerciale(IItaly)	UPS useful to guarantee electric power supplying in case of few hours PC electricity problems.
1 x	Laptop Keypad	Tucano portable Keypad	Tucano (Italy)	Laptop keypad used to let customer's insert PIN to complete RFID tag payments.

Table 6.3. Bar payment module and RIGEL hardware components

6.5.4 Turnstile management Module

Introduction

The turnstile management module is a subsystem used to manage the customers' access procedure. It is based on the usage of one or more turnstiles with integrated RFID reader, connected to the same network, by what reading customers' RFID tags. After appropriate checks on customer's subscription validity, it allows or denies customers' access. Turnstile management module is ideated to be always active when system is running. It means it should be started automatically at the system activation without requiring the user interaction. As already mentioned in its design paragraphs we decided to develop it as a Service Application that runs on the Server containing DB.

Working structure

The turnstile management module runs as a Service application on the DB server, started automatically at the Windows Server bootstrap. It communicates to the turnstiles through the local network since these integrate and RFID reader coupled to an Ethernet network

interface. This device allows to pack the tags' UIDs in UDP messages to send them through the network to the turnstile management module. Every time the turnstile module receives an UDP message from a particular turnstile, it extracts the UID, retrieves the customer's information related to the RFID tag read and, after the appropriate checks on the his/her access rights, it answers to the same turnstile communicating to allow or deny the access. Every access is registered to the database to let management create future reports to consult accesses tendencies.

Development hotspots

We chose to deploy turnstiles that can be connected to the local network to let the turnstile management module to communicate with them by socket communication. They integrate therefore a device characterized by the RFID reader and an Ethernet interface that packs RFID UID read into UDP packets. Every time a customer passes his/her RFID tag, the RFID reader extracts the related UID passing it to the Ethernet chip that packs it finally in a special UDP message to send it to a particular address, the turnstile management module, through the network. It means that turnstiles and turnstile management module should have an IP address set during the system installation and configuration. Turnstiles require also the setting of other parameters like TTL, gateway address, UDP port etc. . Since packed by a chip, UDP packets are created according to a proprietary format. Therefore, it was necessary to use the Ethernet interface proprietary library provided for free.

Code structure

Turnstile management module was developed as a Windows Service application that starts automatically at the OS bootstrap. It is characterized by a class named TurnstileService that contains the common Service applications methods, ONStart() and ONStop(), and a method called TxpLinkOnMessageRecv() that is registered to the message received handler of a TraxLink object. This TraxLink class is contained in the TraxIPLink library provided with the turnstiles to realize the communication infrastructure. Once received a message, TxpLinkOnMessageRecv() is called executing the following logic steps :

- Tag serial extraction, the UID read is extracted from the UDP message
- Customers data retrieval, all the customers information useful to check access rights validity are retrieved.
- Access confirmation, in case of customer's valid access rights, it sends to the turnstile that sent the access request an opening command to let customer enter the venue.

Hardware Deployment

The Turnstile management module working requirements, determined partially at Design phase, and its functionalities implemented and mostly illustrated in this chapter, conducted us in identifying and choosing the devices useful to allow this subsystem to run.

Considering it runs on the DB server, no PC or laptop was necessary to be bought. In the table below we list all the internal components chosen for Turnstile management module running, reporting quantity, device class, model name, the related supplier and a brief component description.

Q.ty	Device category	Model name	Supplier	Description
2 x	Turnstile with RFID reader integrated	MASTER INOX turnstile	Codarini (Italy)	Three arms INOX turnstile with arrow leds and acoustic signal.
2 x	Access controller	AXESS NEO-MAX 914.013.20, with RS232 interface	Codarini (Italy)	Controller Card to manage RFID reader and turnstile opening mechanism.
2 x	RFID reader	AXESS RFID ISO14443 904.201.14	Codarini (Italy)	ISO 14443 reader to read automatically UID RFID tags.
2 x	Ethernet network interface	AXESS ETHER-LITE 983.010.20	Codarini (Italy)	Ethernet network interface card to convert RS232 to Ethernet data flow. It packs/unpacks NEOMAX RS232 messages into UDP messages to allow software modules connected to an Ethernet network to interact with turnstiles.

Table 6.4. Turnstile management module hardware components

6.5.5 IT Kiosk

Introduction

In the development phase, due to deadline issues and design considerations, both the functionalities and the design of this module were affected by some modifications. In particular, the most important changes happened in the design part: instead of developing it as a web application, we decided to use the windows forms as we did for the other modules. This choice was influenced by some reasons:

- First of all, the knowledge acquired in the development of the Box Office module and of the Remote Interface module, which were developed before the IT Kiosk, convinced us that it would have been more immediate to start a development with a well-known paradigm rather than starting a new one;

- Second, some of the non functional requirements we had to deal with, such as automatic logout and denial of the usage of windows special keys (e.g. Alt+TAB, CTRL+ALT+DEL) resulted to be easier to be developed together with windows forms than ASP.NET web application.

The two reasons above explained, made us have a research on the main practical differences between windows forms applications and web applications. We list these features in the table below indicated:

	Windows Forms	ASP.NET Web Application
User Interfaces, data binding etc.	Easy to build	Difficult to build
Deployment and Maintenance	Complex. New versions of assemblies, configuration files, and other required files must be deployed on all client machines. Usually user interaction required.	Easy. Need to deploy assemblies and configuration files on the server only. Transparent to the client
Performance	Faster	Slower
Robustness and Reliability	One client machine goes down, other users are still live.	Usually web servers are never down. However if the server goes down, all users are affected.
Network Congestion	Depending on the data transfer and connections made to the server from various clients.	Depends
Resources	Runs on the client machine.	Runs on a Web server.
Catastrophic failure	User interaction required.	Usually user interaction not required.
Framework dependency	All client machines have to install required versions of .NET framework and other required libraries.	Only server needs to have .NET framework and other required libraries.

Table 6.5. Windows forms versus ASP.NET web application

The reasons above therefore convinced us to change the design decisions and to use Windows Forms. About the functionalities, some of them were not developed, leaving them as a future possible development. Let's remember what functionalities had to be created in this module:

- Read news and announcements published by the company.

- Buy services.
- Subscribe to entertainment activities.
- Handle the functionalities related to visualization and modification of data related to previous acquired services and activities
- Recharge the personal account
- Handle the personal account thanks to transactions visualizations
- Handle the personal data (modify personal informations)
- Modify the personal password related to the wristband.

The functionalities regarding the purchase of services and activities and of the visualization of news were not implemented. As the Beach Resort is still brand new, the company itself is still working in the marketing related to the type of services and the activities to insert in the structure. These functionalities will come out later as the company will have a clear vision of what kind of services and activities will be offered to their customers.

The graphical organization, due to the limited functionalities developed, resulted to be pretty simple. That was anyway a requirement as this module, whose usability as we already saw is one of the most important constraint, works in a machine with a touch screen device. Buttons and labels are therefore big and easily visualizable: that would have been a big issue if the functionalities to represent were numerous.

Let's analyze now how this module was organized, first in a graphical way, then in the choice of programming structure.

Interaction organization

As we already saw from the previous chapter, this module has 2 levels of authentication: one is the RFID tag autentication and the next is a password authentication. With the first one it's possible to access to functionalities which doesn't contain any personal information and which doesn't raise any privacy issue, the second is used instead to access to sections which contains delicate information about the customer's profile. The application is organized in a full screen mode, so that the customer cannot access to the operative system's functionalities: this is why we had to disable all the keys that could be used to close the application or to switch to another application running in the machine. The application can be closed only if accessed with a special administrator RFID wristband which is kept in the Box Office and therefore not accessible to the normal customer.

First screen that the customer sees is therefore a screen with no buttons that asks the customer to touch the screen and to put it's RFID tag near to the reader. If the tag is recognized and it's linked to a customer who's got a valid subscription, a new window with the available functionalities is showed:

- Recharge account (No password required)

- Handle profile (Password required)
- Handle account (Password required)
- Change password (Password required)



Figure 6.15. The IT Kiosk main form

When touching the button related to the *recharge account* functionality, a new windows is automatically opened; otherwise, a login popup with a password label is shown. If the user inserts the right password, the window relative to the functionality is opened and the user is logged in till a logout is made by the user or a timeout makes it automatically.

Code structure

Since this module was designed to follow the 3-layered architecture, it was organized in the following way:

- A visual studio project contains all the windows forms composing the module
- A class library project called CustomComponents contains the business layer: all the algorithms and the libraries to connect to the hardware devices are contained in this library. This class library project includes also all the libraries related to the data layer.

Three controllers were created in the business layer: one that contains all the functionalities related to the customer (CustomerCtrl), one that contains all the functionalities related to the accounts of the customer (AccountCtrl) and one that controls the functionalities and the machine state of the cash acceptor (CashAcceptorCtrl). Later in this paragraph the CashController ctrl will be analyzed in details.

Development Hot Spots

Low level keyboard hooks As we said before, when developing this application we had to face out the problem of avoiding that a customer who's some computer science knowledge could close the application or pass to another process with a clear problem for the security of the whole system itself. As the application is full screen, and does not contain any exit button, it is not possible to exit the application from the graphic side; but an expert user could easily use the special key combination such as ALT+TAB or CTRL+ALT+CANC to close the ITKiosk application and to have access to the machine where it's installed. To solve this problem, we decided to use Windows Hooks. In Win32, a hook is a mechanism by which a user-defined function can intercept a handful of system events before they reach an application. Typical events that hooks can intercept include messages, mouse and **keyboard actions**, and opening and closing of windows. The hook can act on events and modify the actual flow of code. A hook is ultimately a callback function that applications register with a particular system event. Each hook function is responsible for passing control to the next one in the chain when finished. If the filter function doesn't call the next hook, the operating system takes control of the flow and considers the event-hooking phase completed. Applications register functions to hook certain events using API functions like SetWindowsHookEx, which adds a filter function to a particular event's hook chain, and UnhookWindowsHookEx, which removes it. A fundamental aspect of hooks is their scope. Normally, hooks may have either system or thread scope. A few, however, can only have system scope. When a hook works at the thread level, it can only trap events generated within that thread. Similarly, a systemwide mouse hook gets called whenever the user moves the mouse, regardless of the particular thread that handles the event. A system-scoped hook is called to handle the event for all the currently running threads. This poses a precise context problem.

In our application we used the thread hooks to intercept the following combinations:

- ALT+F4;
- CTRL+ALT+DEL;
- CTRL+ESC;
- ALT+TAB

A custom function was used instead to disable the "Windows" button.

Hardware Deployment

The IT Kiosk module working requirements, determined partially at Design phase, and its functionalities implemented and mostly illustrated in this chapter, conducted us in identifying and choosing the devices useful to allow this subsystem to run.

After a careful research on ready to use Informative Kiosks, in what deploy our IT Kiosk modules, we decided to design and build by ourselves a new one. Comparing the costs to buy an already designed and ready to use Kiosk and the costs to create a new one, we

noticed we could spend half the necessary money.

The IT Kiosk infrastructure design required a particular attention on air flow and humidity problems due to the beach resort climatic features. The hotness and humid-salty air could damage internal IT Kiosk electronic devices. Therefore an airflow system composed of two fans were placed inside and silicon closures made when necessary.

In the table below we list all the internal components chosen for IT Kiosk module running, reporting quantity, device class, model name, the related supplier and a brief component description.

6.6 Networking

In the table 6.7 we list the network components we used to realize a local network for ITBeach. The Ethernet local network relaized is based on a central server containing the database to what is possible to access thorugh SQL Server 2005 EXPRESS service running on it. The switch represents the center of the star-topology chosen to connect all the devices. The most part of devices deploying the subsystems previously designed are connected to the local network through LAN cables, while wireless routers are useful to allow PDA devices deploying PDA subsystems to connect to the local network. A gateway with firewall funcitons integrated can be used for future developmens requiring Internet connectivity.

Q.ty	Device category	Model name	Supplier	Description
1 x	Desktop PC	HP KV364ET	Informatica Commerciale (Italy)	Desktop PC characterized by the E7400 Intel Pentium dual core processor(2,66 GHz/1M); 3 GB DDR2 RAM; 250 GB HD; DVD reader; shared memory graphic card; 1Gb Ethernet Network interface; Windows XP Home Edition OS.
1 x	Touch screen	LG 1730 SF, 17" touch screen	Informatica commerciale (Italy)	Touch screen monitor useful to let customer interact with IT Kiosk software.
1 x	RFID reader	Arygon, ISO 14443-NFC compliant , USB Desktop reader	IDTronic (Germany)	USB RFID desktop reader compliant with NFC and ISO 14443A standards. It is used to read MIFARE 1K RFID tags worn by customers to access the module functionalities.
1 x	Receipt printer	Custom TG 2460 II, 60 mm thermal printer	Custom (Italy)	60 mm Receipt printer used to print no-fiscal receipt related to customers' credit recharges.
1 x	Cash acceptor	VTI Matrix RS232 cash inserter	Custom (Italy)	Cash acceptor device useful to let customer recharging his/her own electronic account. 5,10,20 euro bills can be inserted.
1 x	Uninterruptible Power System(UPS)	APC, Schneider electric	Informatica Commerciale (Italy)	UPS useful to guarantee electric power supplying in case of few hours PC electricity problems.
1 x	WaterProof Keypad	TATSUYAMA waterproof keypad	Informatica Commerciale (Italy)	This keypad lets users to modify PIN or profile's info. We chose a waterproof one considering this open air application context(beach resort).

Table 6.6. ITKiosk hardware components

Quantity	Component	Description
	LAN cables	Necessary to create the whole local network infrastructure
1 x	Switch 16 ports	Necessary to create the whole local network infrastructure. The LAN cable connected to each component (turnstile, PCs, wireless routers) will be connected to a switch port
2 x	Wireless Routers with access list and Firewall	Covering the entire structure's area, they provide PDA devices connection to the local network
1 x	Gateway with Firewall	It is intended to be the edge point of the network by what accessing to networks external to the local one(e.g. Internet)
1 x	Server	It's the machine containing the DB for the IT Beach system. Its DB can be queried by every allowed device deploying IT Beach modules

Table 6.7. Network hardware components

Part III

Analysis of results and project closure

Chapter 7

Analysis of results

7.1 Overview

In this chapter we'll analyze the important data it's been collected in these 2 months in the central database. It's extremely interesting to examine this information because we can have very useful hints about how the software is accepted by the customers, about the learning curves of the employees, and about data that can be used from the stakeholders for marketing purpose. The data we analyzed was based on a relatively short period (1st may 2009 - 21 June 2009), but as we'll see in the following paragraphs it was possible to extrapolate some interesting statistic.

7.2 Bar payment analysis

In this section we're going to analyze the transactions made at the BAR. As this functionality started on the 20th of may, the analysis is based on a shorter period. As it happens for every introduction of a new technology, it takes time for the customers to become familiar with it. **Diffusion** is the process by which a new idea or new product is accepted by the market. The rate of diffusion is the speed that the new idea spreads from one consumer to the next. From the chart we made from the data extrapolated from the database, we can make a study about this service's diffusion. Let's examine it:

In the chart we can find two different information analyzed every week:

- The number of transactions happened every week
- The amount of money spent within these transactions

In the first information, we can see an expected result. In fact, transactions are growing every week in a pretty linear way, even if in the last week we could notice a bigger growing rate. As it's expected, after a slow start the diffusion of a technology starts growing in an exponential way. After the so called "innovators" and "early adopters" take possess of the technology, more and more people starts to be familiar with it and contribute to create trustiness between the whole base of customers. We can see a typical innovation

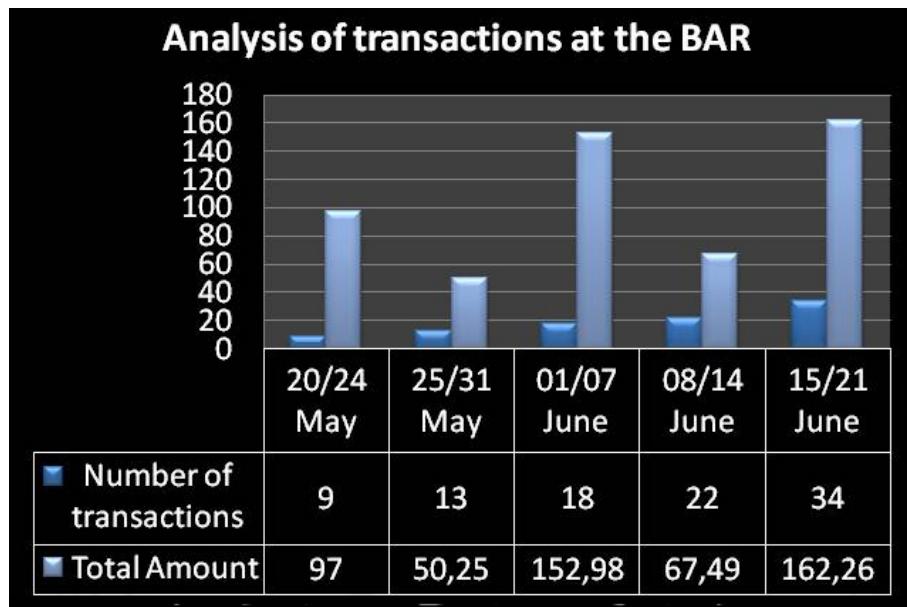


Figure 7.1. BAR payment analysis (the amount is indicated in euro)

curve that explains this process in the figure below (more information about this process can be found at Value Based Management website)¹:

What we expect now, is the exponential growing. This analysis will be possible to be analyzed in next months. However, the stakeholders could help the diffusion of this technology by providing some marketing strategy to involve the customer to use the virtual account rather than money for example by applying discounts.

Second information provided in the chart is not giving any useful statistic at the moment. Inf act, the money spent is completely random and not connected to the number of the transactions. As we can see in the chart, for example, in the 8/14 june week a number of 22 transactions were made, but the total amount of money spent for these transaction was 64,49 euro. The previous week, with a smaller number of transactions (18) the total amount of money spent was 152,98 euro, which is about 300% respect to the following week. For this reason, at this moment with the small amount of data related to this information it doesn't result to be particularly interesting. Most probably with more data available it will be more probable to extrapolate some interesting information also from this data.

7.3 Virtual account recharge analysis

The second analysis is dedicated to the functionality of recharging the virtual account used for the electronic payment inside the structure through the RFID tag. This analysis is strictly connected with the first one, as the virtual account at this moment can be

¹http://www.valuebasedmanagement.net/methods_rogers_innovation_adoption_curve.html

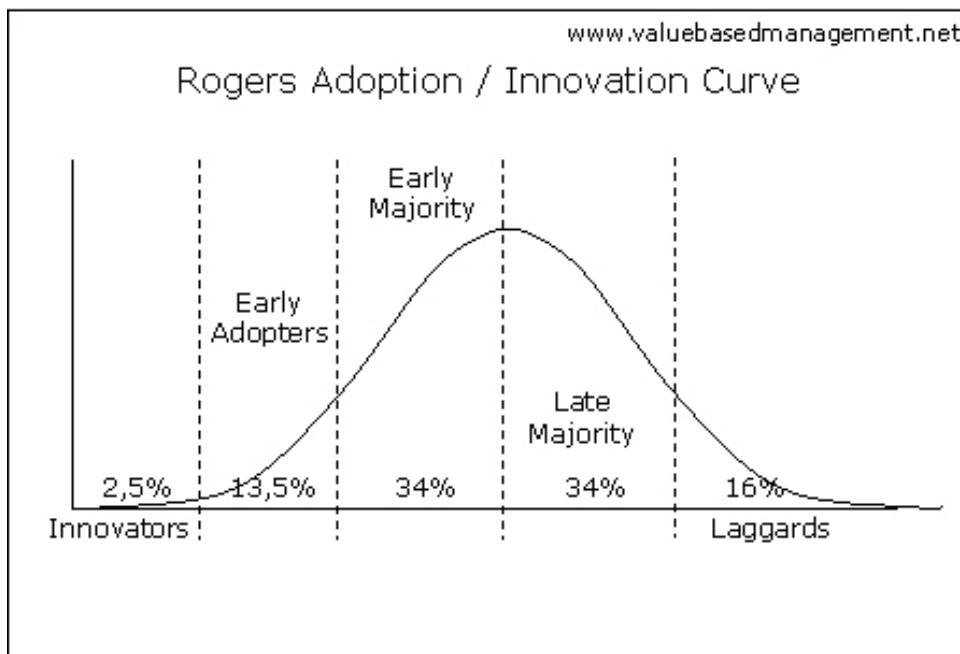


Figure 7.2. Innovation adoption curve

used only for the BAR payments. For this reason, all the concepts about the innovation diffusion curve explained in the previous paragraph can be applied also for this data. What makes this analysis interesting it's the fact that this functionality can be made both at the Box Office and at the IT Kiosk. It's interesting to notice the connection between these two sources. Let's analyze the chart in the picture below:

This chart deals with 4 different variables, updated every week:

- Number of recharges of the virtual account through the Box Office
- Total amount of recharges of the virtual accounts through the Box Office
- Number of recharges of the virtual account through the IT Kiosk
- Total amount of recharges of the virtual accounts through the IT Kiosk

From what we analyzed in the previous chapter, we could expect the same behavior also in the adoption of this technology. And in general, considering the operation of recharging the personal account this is true. But we also noticed an interesting difference between the operations in the two modules.

In the first three weeks, the recharge operations made at the IT Kiosk module were almost none: 1 the first week, 0 the second week and 3 the third week. If we analyze the same data for the same operation at the Box Office we can notice that there were 8 operations the first week, 5 the second week and 8 again the third week. Before continuing,

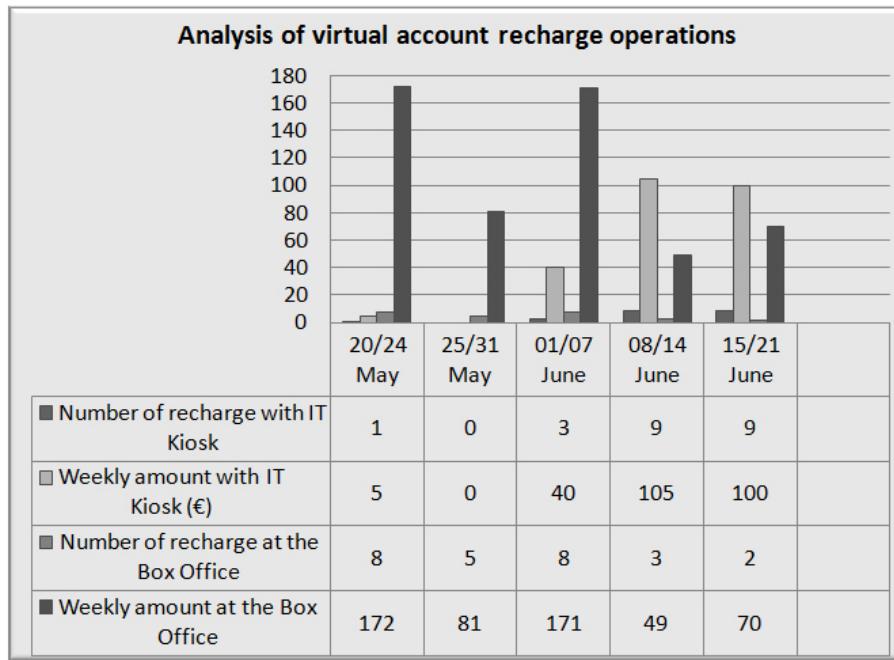


Figure 7.3. Virtual account recharge chart

it's important to say that these values depends also on the number of customers present inside the structure for each week, and this data is influenced by external factors such as weather, national not-working days and so on. For this reason some result can be a bit different from the expected :for example the fact that the operations in the second week are less than first and third wouldn't make any sense. But if we consider that in the second week from the introduction of the functionality the weather was not perfect, then we can understand how in a beach resort there were less people accessing the structure and therefore available for this type of operations. By the way, the complete lack of recharge operations at the IT Kiosk can be explained mostly with lack of confidence of the user towards the machine: while in the Box Office the customer just had to give money to the employee who would perform the requested operation, in the IT Kiosk the he/she would have had to perform the operation by himself/herself using its RFID tag and the cash acceptor present in the automatic Kiosk. So, even if the Box Office is possibly less uncomfortable as it could have a queue of people who want to check in or check out, the customer preferred to wait and to perform the operation there. The last two weeks of analysis shows something extremely interesting:

- An increase in the recharge operations made at the IT Kiosk.
- A decrease in the recharge operations made at the Box Office.

As we can see from the chart, in fact, the recharge operations at the Box Office in the last 2 weeks are respectively 3 and 2, in the IT Kiosk are 9 for both weeks. This trend

can be explained by the concept of learning curve: the term learning curve refers to a graphical representation of the “average” rate of learning for an activity or tool. It can represent at a glance the initial difficulty of learning something and, to an extent, how much there is to learn after initial familiarity.

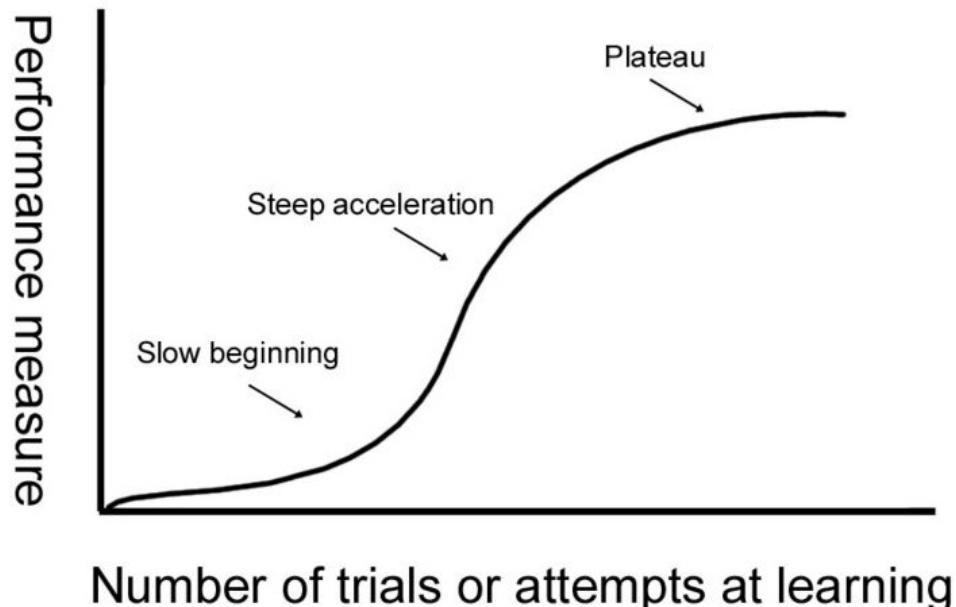


Figure 7.4. The learning curve

As the IT Kiosk is a pretty simple application, it presents a relatively small phase of “slow beginning” therefore the learning curve for this application is pretty small and steep. This is why, after a slow start, customers started to use the IT Kiosk instead of the Box Office to recharge their account: this made the operation faster as they didn’t have to wait any queue to perform it. The other data, that is to say amount of money recharged, shows us another interesting issue: the total amount of money respect to the number of recharges performed is much bigger at the Box Office than at the IT Kiosk. This can be explained in two ways:

1. People does not trust “machines”, so it’s more difficult that they rely a big sum of money to it as they think that something can go wrong and they could eventually lose the money. On the opposite, there is less distrust on relying money to a human.
2. The cash acceptor accepts only small banknotes (5-10-20 euro). For this reason one can decide to recharge the account time to time instead of putting more money on the account.

7.4 Check-in execution mean time

The check in procedure execution mean time, intended to be the average time in what the employee is able to execute a new subscription or a daily access procedure, is calculated in minutes, as the ratio between the number of subscriptions made over a time interval. The analysis of this data allows us to understand how efficient is the clients acceptance procedure. In fact, it must be simple in order to allow a quick queue management. Long time attendances could annoy customers causing the related inconveniences. Ideally, the time needed to complete a subscription should be the same or less than the one required before the introduction of the system. We have to arrive, as closer as possible to this goal.

To obtain significant and useful analysis results, we should observe really crowded days and "hot"(crowded) time intervals of accesses. Therefore, considering that May and June are not so crowded periods since it's not summer yet, we decided to observe only sunny Sundays of the time period ranging 10th of may and 22nd of June. We excluded only the 31st of May cause it was a cloudy day and few subscriptions were made.

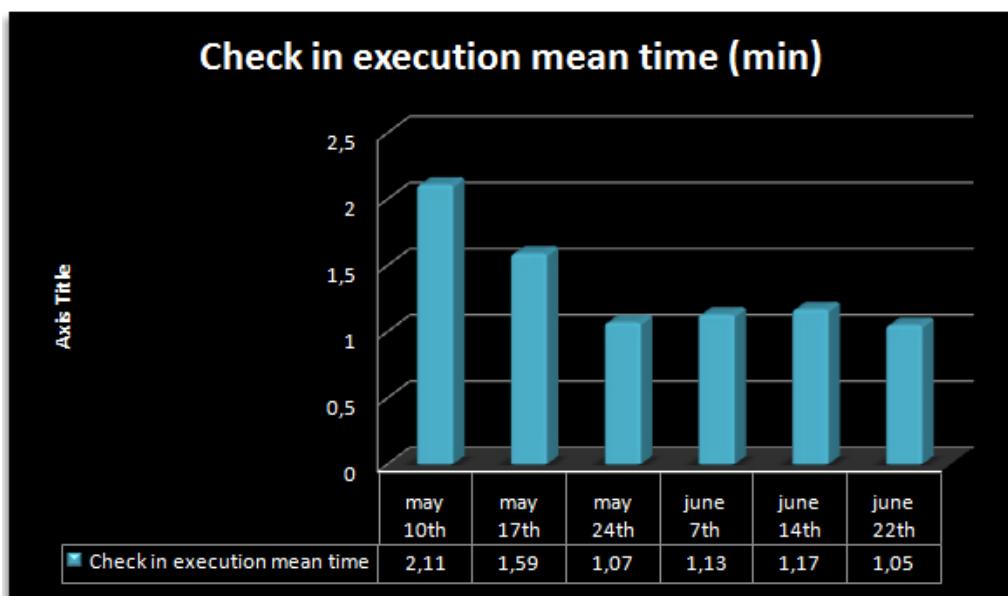


Figure 7.5. Check-in execution mean time chart

Let's start our analysis. As we can observe from the graph above, the execution mean time of the subscription procedure decreased rapidly along the first three weeks. From a 2.11 min in the second week(may 10th) of may we fell to 1.59 min per procedure in the third week(may 17th). We can immediately notice that after just 7 days of practice, employee results more confident with the application usage. It seems that resistance to change and usage fears are going to be defeated now. There is big a gap of 30 seconds between these two weeks, but although a good improvement was done, 1 minute and 35 seconds to complete this task is still a lot. In the fourth week(may 24th) we can observe

a new rapid decrease; we reached just 1,07 min per procedure, reducing again almost 30 seconds. After this last rapid decrease the curve seems maintaining a flat proceeding(1.13 - 1.17 - 1.05) making us thinking that new consistent improvements are difficult to reach. Nowadays we reached a good check in procedure execution mean time since it oscillates between 1.05 and 1.17 that was halved compared to the 10th of may. It demonstrates that the employee had to practice for at least 10 days before reaching a high confidence in using the system. What we expect now, is to maintain more or less this curve trend, even if, to be more accurate, we should observe this curve again in the next months(July and August) in what we expect more people.

7.5 Tag losses trend

The graph below has the objective of showing the trend of the tags losses in the period ranging 1st of may and 22nd of June. The high delivery time and investment costs related to RFID tags purchases require a careful analysis on the tag loss trend during this first period of the system execution. Every customer has to have the RFID tag since it's necessary to let the system running correctly. Bar payments, Kiosk access, Turnstile accesses functionalities are based on its usage. A lack of tag's would cause the incorrect and not complete running of the system. A necessary analysis of the tag losses or damages has to be done to anticipate dangerous decreases of their amount to immediately intervene with new orders. As already mentioned before, we decided to buy 1200 fashionable RFID wristbands from a Chinese Supplier since it proposed a good quality-price ratio compared to European suppliers. We estimated to buy 1200 RFID wristbands basing on the number of emplacements offered by the structure. Every emplacement can host maximum three adults and two children. Considering of having 240 emplacements, that every emplacement hosts in average two adults and a child or three adults, considering that children don't receive RFID wristbands (they access for free), we calculated to buy 600 RFID wristbands, considering an average amount of customers for each emplacement occupation equal to 2.5 ($600 = 2.5 \times 240$). To be sure to face tag losses and damage events, we doubled the amount reaching 1200 items. This quantity was considered sufficient to complete the whole season without LACK OF TAGS.

During these last weekends, we noticed a rapid decrease of the amount of our RFID tags at the Box Office emplacement. This phenomenon required a deeper analysis to understand how many RFID tags were lost or damaged and if was necessary to make new orders. Therefore we created a chart gathering information on subscriptions done and tags lost, from all the weeks (Monday-Sunday) ranging the 1st of may and the 22nd of June. Let's examine it. In the chart we can find two bars, that represent different data analyzed every week:

- The number of tags lost, multiplied for a scale factor equal to 10(to let its bars rendering better on the scale of subscriptions amount)
- The number of subscriptions done

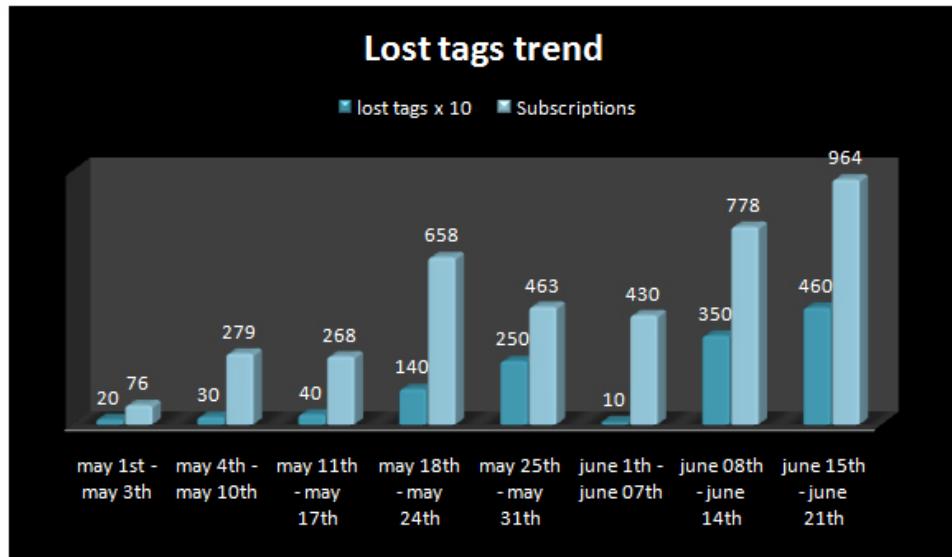


Figure 7.6. Lost tags trend chart

The trend shown by the bars confirms what expected: the number of subscriptions made, grew during the weeks since we are getting closer to the summer season, but together with the raising of tag lost number. This last trend was quite obvious, since the increasing of the subscriptions completed imply the natural increasing of the probability to loose or damage assigned tags. As we can observe from the chart, during the first three weeks the number of tag losses was really small, equal to $9(20+30+40=90)$ divided 10, the scale factor). But from the 4th (may 18-24) week to the 8th (June 15-21), the losses raised worryingly reaching the amount of 35 and 46 tags lost in the last two weeks. We should not consider bars of the week June 1-7, because system malfunctions occurred and no RFID tags were used. Considering the almost flat proceeding of the curve in these last two weeks , considering we are getting closer to the summer , the number of subscriptions will increase and consequently the number of tag losses. Just looking at the last two weeks, assuming that a number of 964 subscriptions made is almost an high amount per week, we could expect to lose during the next weeks a number of tags ranging 30 and 50 units or even more. This observation is quite worryingly since we don't have a huge number of tags and we didn't expect so high losses percentages at the season start. Let's further analysis how many tags we've lost for every subscription completed. Calculating the ratio we obtain the chart below.

Let's make some calculations to understand if we can arrive to the end of the season using only the 1200 RFID wristbands we ordered in April , before the system running:

Tags lost till 23rd of June = 130 tags Tags given to employees= about 53 Tags given to periodic subscribers= about 432

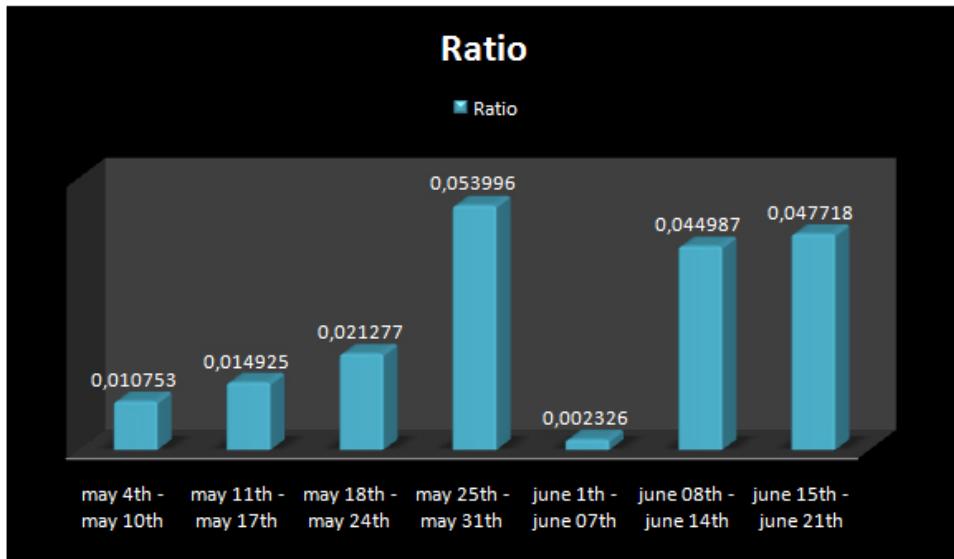


Figure 7.7. Tag loss ratio

1st Result Available tags at Box Office Emplacement for daily subscribers at morning= about 585 tags Considering that every morning the structure rent all its daily emplacements that are equal to 128, considering that 2.5 adults in average occupy one emplacement, we need every day 320 tags available.

2nd Result Average of not used tags at Box Office after rent all the daily emplacements during the morning= $265(585 - 320)$ Do we expect to maintain constant this RFID tag amount, till the end of the season, considering the losses percentages of these last two weeks? Absolutely not! Considering just the average number of the tags lost the last two weeks, loosing 40,5 ($35+46 / 2$) per week would mean set to 0, just after 6 weeks, the amount of not used tags after rented all emplacements every day. It means that after the 26th of July we will not have enough wristbands to let customer's occupy all the structure's daily emplacements. And what happen if a periodic customer loses his/her RFID tag? Since these not used tags ended, he/she cannot have a new one.

This is a situation that has to be avoided in any case, because it causes big issues in terms of:

- Money loss, in theory, all daily emplacements cannot be rented because enough tags are available to let every customer having access at turnstiles.
- Daily customers' complaints, since the Mondello beach is almost privatized, customers can comfortably access to the sea by the MIIB beach resorts. Denying them the access because of tags missing, means denying them to go to the sea. Obvious complaints will arise.
- Periodic customers' complaints, it's not possible to deny them the access only because there are no available tags to assign. They have already paid for the whole

season.

A new purchase order Considering the previous analysis results, a new and quick order has to be done. Quick because, we risk anyway to arrive to the 26th of July without tags since the time required to product and deliver new RFID tags ranges between 4-5 weeks. The only way of gaining time is trying to reduce the losses percentage increasing the bail price. In this way, we could expect to impact this phenomenon reducing it for more than the 30%.

Chapter 8

Project closure and future development

8.1 Post-completion project overview

The previous chapter provided an overview of the statistics of system functioning for the period May-June. Apart from the analytical data, now that the project is over, it's important to give an overview of it in order to understand:

- To what extent project expected results were achieved;
- Errors and successes on fundamental project activities;
- what went differently than expected in terms of time and budget for completion against initial time and budget estimation;
- what went right beyond expectations;
- what went wrong beyond expectations;
- what were the lesson learnt;

8.1.1 Achievement of project expected results

In chapter 4 we reported what were the expected results in relation with project goals. At this point it's necessary to consider each one of these results and analyze to what extent they can be considered as achieved.

MIIIB's image improvement

The introduction of the system has induced a lot of positive comments among clients and media. The only negative comment we have to report is the one that put in relation the RFID tag with the project of providing prisoners with electronic wristband for geographical positioning. This was due to the terms used to refer to the RFID tag (electronic wristband)

which was the same used in the jails context. As we spread a precise name for it, the "SeaPass", these comments were sensibly reduced. Anyway, since this phenomenon had quite little impact on the general attitude around the project, we consider this result as ACHIEVED.



Figure 8.1. Article on Repubblica newspaper's website

Improvement in quality of services provided to clients.

It's not easy to make an evaluation of this result since the number of services provided to the client is still limited. We'll then make some considerations on the services actually present.

- **Check in procedure:** the check in procedure is now more efficient and effective since the client is able to visualize the resort map and choose the location he/she prefers. The need for continuous via-radio contact between the box office and beach assistants has been eliminated since the locations availability is now provided by the system and not by the beach assistants. The system prints a receipt with the location number on it, so the chance for a human error in writing the number of the assigned location has been eliminated.
- **Access to the structure:** clients who are accessing the structure for the first time often require the assistance of a beach assistant to understand how to perform a correct tag control at the turnstile. This is due to a lack of visual hints for the client on the turnstile or around it. Anyway, once the client is confident with the procedure, the access to the location is no more dependent on the presence of a beach

assistant controlling the receipt. These brings to a better service at the moment of the access to the structure.

- **Purchases at the bar:** since the customer is free to choose what type of payment to use (money or virtual account) to purchase items at the bar, and as we've noticed an ascending climax in the use of this service, we can consider this service as improved.
- **Other services:** the system as it's developed, supports also the selling of services and activities of different types. Anyway, as already mentioned in the previous chapters, these are still not available due to management issues. Therefore it's not possible to analyze the improvement of this section.

As we noticed a general improvement of the services offered to the customers, we can consider this result as **ACHIEVED**.

Automatic and effective access control that prevents abuses

The turnstiles introduction is having a huge (positive) impact on access control. The situation previous to system introduction was based on a receipt control performed by beach assistants when the client was accessing the structure for the first time. Since it was possible to exit the structure and enter again, the control was dependant on this piece of paper (annoying to carry along in a beach context) and therefore more often on beach assistants memory of clients' faces. In the more crowded days it was impossible to keep track of all the people entering the structure. The functioning of the turnstile, as explained in chapter 6, provides a strict, quick and continuous control preventing any abuse. This result is to be considered **ACHIEVED**.

Permeation of a proprietary electronic payment system inside the beach resort

An electronic payment service is already available at the BAR. As evinced from the data analyzed in the previous chapter, after an initial slow start due to customers' distrust, an increase on the usage of this service is taking place. We can thus consider this result as **ACHIEVED**.

Achievement of a stable information basis for short, medium and long term strategic decisions

In developing our project, we made sure to collect every kind of data that could result to be interesting for the strategic decisions of the company. Together with information about customers and subscriptions, we collect data about successful accesses, not successful accesses, customer's opinions about services, system's malfunctioning and much more. As we could already extrapolate interesting analysis from only two months of collecting data, we can consider this result as **ACHIEVED**.

Strict control on beach furniture and services fruition inside the beach resort.

This result is related to the development of a periodic rounds mechanism, supported by technological devices, intended to reveal any abuse inside the structure. This mechanism is been designed inside the PDA Module. Since this module has been designed and not implemented, this result, until now has to be considered **NOT ACHIEVED**.

8.1.2 Errors and successes on fundamental project activities

Feasibility Study

The market analysis that led us to the hypothesis of this project identified a real area of opportunities in the present situation. The Italian environment is still showing a lack of this kind of integrated solutions deploying RFID technology and our solution could be able to fulfill to this need. The feasibility study we arranged didn't miss any of the project deliverables we had to build. Furthermore, another positive aspect in the released feasibility study is the absence of unnecessary deliverables. On the other hand, it probably underestimated the time needed for several project's phases, resulting in a overcrowded list of deliverables for the timing constraints we had (i.e. PDA Module and Advertisement Module). The preliminary phase of our project lacked of a more accurate analysis on RFID devices' and tags market situation. Reaching earlier the awareness on compliance and pricing situation could have brought us to an earlier identification of potential suppliers, leaving us a bigger margin on contracts negotiation and delivery slack time.

Project Planning

Our planning estimations resulted to be quite ineffective with regards to some of the project phases, especially to implementation one. The reason for such poor estimation is connected, first of all, to our little experience on projects of this size. We went through different type of delays, from the Credit Letter requested by the Chinese tag supplier to the difficulties involved in turnstiles configuration. On the other side, the mentioned continuous changes in requirements had a considerable impact on the schedule, especially while in the implementation phase. The biggest change in requirements to report has been the request to provide all Banca Nuova's bank agencies in Palermo with a Remote Box Office Module at the end of February. The delay introduced by this requirement affected all the rest of the project. Since we were afraid that our project wouldn't have been accepted by MIIB's management, we tried to make a perfect estimate without pumping every estimation with an almost natural 20%. Furthermore, future estimations will get more accurate as our competencies and skills in this kind of projects increase. We took the largest advantage from personal skills and nobody had to face any task that another resource would have faced better, thanks to previously acquired competencies or shown predisposition. Even without the mentioned changes in requirements, the release of missing deliverables would have fallen beyond schedule for some delta. The early signs that this would have happened arrived from IT experts and teachers. After a project overview, many of them tried to avoid encouraging our optimistic approach, but we kept on believing that

the project was possible in all its parts. Just in a later moment we understood the reason why those experts were so doubtful about project completion in time. The subsequent lesson learnt is already clear.

Requirements Elicitation and Analysis

Since the project was executed in Tallinn (Estonia), the only way to communicate with MIIB's management and staff was through telephone calls, instant messaging or email exchanges. This cannot for sure be equated to a direct and personal contact with the customer especially during requirements elicitation. This can be considered at the base of the most frustrating part of our project, consisting of a big number of reiterations over requirements elicitation phase. In the worst case, some requirements changed when our team was approaching the release of the deliverable affected by that change. In order to avoid this frustration, it would be much more better to spend some extra effort and time over the initial requirements elicitation, adopting a more structured approach. The direct contact with the customer should be mandatory and all interviews and summits accurately documented and reviewed.

Development

The most convincing method we adopted in this project has been the per-layer responsibilities decomposition adopted during the development phase. This method allowed us to decouple modules components and to work on them separately and in parallel. The Strict Layering model architectural style adopted required two interfaces: between Presentation and Business Layer and between Business and Data Access Layer. Each functionality has been faced with this approach: Presentation Layer responsible and Business Layer responsible had to agree on data and computations needed by the former that should have been provided by the latter; Business Layer responsible had to ask to the Data Access Layer responsible a suitable access to required data, make eventual computations and provide results to the Presentation Layer responsible. This pattern worked out very well and could be easily replicated.

Deliverables release

Two distinct considerations have to be made regarding applicative deliverables and documental ones. For the documental, in fact, the stakeholders who reviewed the deliverables didn't provide much meaningful input for two main reasons (strictly connected): the documents presented were too big (hundreds of pages) and the stakeholders didn't have much experience with the analysis of software-related documents the fact that we were in Estonia for the development of this project made impossible to run some meetings to explain how to interpret the documents, that were sent by emails. For the applicative deliverables, on the contrary, we got a lot of interesting suggestions on how to improve functionalities and user interface.

Other considerations

The small number of human resources imposed us to be project managers, analysts, designers, developers, testers, installers and maintainers. This concentration of responsibilities over only three resources brought to an overwhelming of tasks per resource. In order to comply with timing constraints, we had to renounce to the monitoring and control of some areas of concern in Project Management. The methods and processes involved in these areas of concern were difficult to use, since they had to be executed in parallel with the ones involved in analysis, design, development, testing, installation and maintenance phase. 90% of what went wrong in this project is a consequence of the small number of resources involved. If we could wave a magic wand and change anything about the project, we would add at least a couple of resources to the team, giving them very specific responsibilities most likely concerning implementation issues and vendors/suppliers management. We analyzed the chance to avoid two of the vendors/contractors involved in our project: the tag provider and the Bar/Restaurant software vendor. Both these entities resulted to be just some sort of intermediary between us and goods/software producers. In the former situation, the figure of an intermediary between us and the tag producer could not be avoided, since otherwise our lack in Chinese market and intercontinental business knowledge would have been definitely penalizing. In the latter case, the intermediary resulted to be completely useless and could have been avoided trying to establish a direct contact with software producer.

8.1.3 What were the lesson learnt

After the considerations made in the previous paragraph, we provide here a list of the lesson learnt:

- An early deep market analysis of available products can help to save time in suppliers selection;
- Project management activities are very hard to conciliate with analysis, design and development phase;
- Don't waste time trying to do things "perfect" if you have enough resources to do them "good";
- Always consider an extra time for extra-European suppliers, and not for delivery times -in some cases even smaller than the ones exposed by Italian suppliers- but most of all for the time needed to communicate required specifications and to get the production/assembly started;
- It's better to spend some extra time on requirements elicitation since every change on specifications will have a higher impact on schedule as the project proceeds;
- It's important to have face to face meetings for requirements elicitation instead of distant communications;

- Too big documental deliverables will be hard for the customer to review and feedbacks will be penalized;
- Facing applications development through a per layer decomposition of responsibilities enables parallel work and effective integration;
- Be sure to employ the right number of human resources for a project. A small number of resources can have a big variety of consequences on project outcomes.

8.2 Future developments

We'll now present what future development can be foreseen for the system interested by our project.

8.2.1 Missing modules and functionalities

The priority has to be given to the missing modules for which we already executed requirements analysis and design (PDA module and Advertisement module). The development of the advertisement module will be supported by the information we already collected in these months so we'll be able to understand quickly if those information are suitable for calibrating an "intelligent" advertisement rotation on screens. Some advertisement could be introduced on kiosks and would be targeted on the current logged user.

8.2.2 Data mining

Data mining is the process of extracting hidden patterns from large amount of data. When the actual season will be over, the application of data mining techniques and tools over the collected data could let us discover behavioral trends useful for the management. Further and progressive refining of those techniques and tools could lead to the implementation of a powerful application able to extract a big number of useful information from raw data, otherwise impossible to analyze.

8.2.3 Customer profiling

Data mining concepts are often related to customer profiling issues. A lot of firms, like the MIIB shows a lack of knowledge on their customers. The amount of data collected by our system could be easily integrated with the ones collected by the bar management software. Crossing the information, it would be possible to perform an effective customer profiling in order to achieve higher quality of services provided to clients and higher knowledge on customers' behavior.

8.3 Conclusions

The system implemented has been successfully installed and it's currently functioning in Mondello beach (Palermo). It results to be the first beach in Sicily and among the

firsts in Italy managed with a system deploying RFID technology. Five out of six project goals identified during the planning phase have been achieved. The budget estimation has been quite accurate since it's been underestimated with a discrepancy of 18% compared to effective expenses. Not so bad since nobody in our team had previous experience on feasibility study arrangement and costs estimation. A couple of considerations have to be made on the personal and professional growth that this project has enabled:

- The vast quantity of IT technologies and fields of study involved in this project make it as a convergence point among the several disciplines characterizing the academic path of an engineer in computer science;
- Task and responsibility assignation, together with the need for an approach enabling parallel execution of activities, either in analysis, design and development phase, gave its contribution towards the improvement of our communicative and collaborative skills.

Appendice A

Riassunto in lingua italiana

A.1 Introduzione

Il progetto presentato in queste pagine è stato condotto al fine di completare il percorso specialistico in Ingegneria Informatica al Politecnico di Torino. Esso costituisce la tesi magistrale dei tre candidati riportati nell'intestazione.

A.1.1 Scopo della tesi

Il lavoro di tesi è consistito nella realizzazione, dall'idea all'effettiva installazione, di un sistema informativo (SI) che sfruttasse la tecnologia RFID a supporto della gestione di strutture ricettive e commerciali (ad es. parchi divertimento, poli fieristici, impianti sciistici etc). Il sistema ha mirato ad automatizzare i processi caratteristici della gestione di questo tipo di strutture ed a predisporre un framework su cui basare scelte manageriali di breve, medio e lungo termine. Un dispositivo indossabile (tag) basato su tecnologia RFID, ha costituito lo strumento chiave per l'identificazione di visitatori/clienti, che è stata pertanto fattore abilitante per tutte le funzionalità previste dal sistema realizzato. L'obiettivo principale è stato quello di maturare le competenze necessarie all'integrazione di soluzioni informatiche ad hoc, nei confronti della quale il mercato nazionale ed internazionale mostra le migliori possibilità di successo. Tra gli obiettivi della tesi è stata individuata inoltre la strutturazione del progetto in modo ingegneristico e l'utilizzo di una metodologia di sviluppo che seguisse gli standard e le 'best practice' riconosciute dalla comunità scientifica internazionale per la realizzazione di sistemi software.

A.1.2 Contesto della tesi

La possibilità per lo sviluppo di un sistema informativo che sfrutti la tecnologia RFID deriva da diversi fattori:

- Decremento sostanziale nei costi di accesso alla tecnologia;
- Larga convergenza su una serie di standard internazionali;

- Possibilità di un incremento del proprio business legato all'introduzione della tecnologia RFID;
- Alto livello di affidabilità, robustezza e durabilità dei componenti RFID in generale.

Le conoscenze acquisite finora in questo contesto tecnologico relativamente nuovo, stanno gettando le basi per piccoli, medi e grandi integratori di sistema interessati a coglierne tutte le potenzialità. Allo stesso tempo, questi potenziali, assieme ad una sensibile riduzione dell'investimento richiesto per l'accesso alla tecnologia in questione, spingono un numero crescente di aziende verso l'introduzione di soluzioni tecnologiche in grado di sfruttarli. Con la crescita delle applicazioni basate su RFID a supporto delle filiere produttive (supply chains), della gestione di magazzino e tracciatura di beni, il cosiddetto *internet delle cose* sta diventando una realtà. Mentre questa tendenza sta portando le soluzioni a supporto della logistica, dei trasporti e dell'identificazione di animali verso un grado di relativa maturità, lo stesso non può dirsi per le applicazioni che presuppongono l'identificazione di persone. Inoltre, il grado di diffusione dell'RFID nella vita di tutti i giorni è condizionato da una serie di critiche e scetticismi circa la sicurezza, la privacy e la salute delle persone. Ecco perchè nella maggior parte dei paesi sviluppati si sta assistendo ad un processo di regolamentazione governativa, in grado di fornire i mezzi per un'introduzione controllata di sistemi RFID. Ciò nonostante, il grado di diffusione di sistemi basati su riconoscimento di persone tramite RFID è piuttosto basso. Questo non può che far rivolgere lo sguardo di ricercatori e compagnie verso quei paesi che si sono finora distinti per avere in qualche modo anticipato i tempi, intuendo prima di altri le potenzialità della tecnologia in esame ed investendo maggiormente in essa. L'Estonia può essere certamente annoverata tra questi. L'analisi, il design e l'implementazione del nostro sistema sono stati svolti a Tallinn (capitale dell'Estonia), ed abbiamo potuto contare sulla collaborazione della Eliko Competence Center, un'azienda di ricerca e sviluppo fondata dalla Tallinn Technical University che collabora con diverse università della scandinavia e dei paesi baltici.

A.1.3 L'idea ed il case study

Da un punto di vista generale, la nostra idea consiste nello sviluppo di un SI capace di automatizzare, attraverso l'uso della tecnologia RFID, tutte le procedure coinvolte nella gestione di strutture che condividano le seguenti caratteristiche:

1. uno o più punti di accesso controllati;
2. uno o più banchi informativi;
3. uno o più punti vendita (bar, boutique, ..);
4. una o più attività svolte all'interno;
5. attrezzatura ed altri oggetti che sia necessario controllare (attrezzatura sportiva, armadietti, componenti d'arredo, ...).

Al fine di ottenere un’efficace analisi dei requisiti che fosse il più possibile reale, nell’ambito del lavoro di tesi e’ stato possibile collaborare con la Mondello Immobiliare Italo Belga SA (per brevità MIIB), una società impegnata nell’amministrazione della spiaggia di Mondello (Palermo) che da pochi anni è determinata in una riconversione di parte del proprio business verso la gestione di spiagge attrezzate.

A.1.4 Approccio al progetto

Il progetto è stato affrontato seguendo un modello di sviluppo largamente diffuso e riconosciuto che prevede le seguenti fasi: studio di fattibilità, pianificazione del progetto, raccolta dei requisiti, analisi dei requisiti, progettazione della soluzione, implementazione, testing, installazione, chiusura del progetto, manutenzione del sistema.

A questo punto è necessario fare una breve spiegazione della tecnologia RFID che ne faccia capire i connotati e le opportunità derivanti dalla sua introduzione.

A.2 Stato dell’arte

Nel precedente capitolo abbiamo introdotto il progetto specificandone ambito e scopi; adesso, partendo da questa base, possiamo procedere e analizzare la tecnologia e l’ambiente per il suo sviluppo. Al fine di essere valido, uno dei requisiti essenziali per la buona riuscita di un progetto è l’utilizzo di standard e metodologie moderne ed universalmente riconosciute per la sua progettazione ed il suo sviluppo. Per tale motivo, abbiamo deciso di creare un capitolo introduttivo sulle tecnologie RFID e sul suo stato dell’arte. I contenuti di questo capitolo includono la definizione della tecnologia RFID, le principali applicazioni ed un piccolo approfondimento su questioni relative a privacy e sicurezza.

A.2.1 Perchè RFID?

Nelle ultime decadi, la tecnologia codice a barre è divenuto lo standard dominante per l’identificazione degli oggetti. Nonostante le sue caratteristiche di semplicità di utilizzo e di bassi costi, le limitazioni di tale tecnologia con l’evolversi dell’informatica hanno portato alla ricerca di qualcosa che offrisse un contesto di funzionamento più ampio per l’identificazione degli oggetti. La soluzione a tale problema è stata trovata nelle *onde radio*. Nonostante le buone qualità di questa tecnologia fossero ben conosciute in numerose applicazioni quali radio e televisione, il costo eccessivamente alto e la non completa conoscenza della tecnologia ha portato ad un rallentamento dello sviluppo di una tecnologia a onde radio per l’identificazione degli oggetti. Una svolta si è avuta all’inizio degli anni 70 con il brevetto di Charles Walton: la sua ‘chiave elettronica’ che sfruttava la tecnologia ad onde radio può essere considerata de facto come lo stereotipo del tag RFID: Radio Frequency IDentification.

A.2.2 Il sistema RFID

Un sistema RFID è composto principalmente da tre elementi:

- Un tag o trasponder
- Un lettore
- Un sistema software che utilizza le componenti sopra indicate (in alcune applicazioni può anche essere un componente hardware)

Vediamo come questi elementi sono interconnessi nel funzionamento di un sistema RFID.

Il tag

Ogni oggetto che deve essere identificato in un sistema RFID, deve essere associato con un tag: pertanto questo può essere definito come quel componente che associato ad un oggetto (può essere anche una persona od animale) consente la sua identificazione univoca senza un contatto diretto, mandando un segnale in risposta ad un comando ricevuto dal lettore. Un tag è internamente composto da:

- Una memoria non volatile che contiene l'identificativo ed eventualmente altri dati scrivibili
- Un microprocessore che contiene l'intelligenza del device
- Un'antenna che viene usata sia per la comunicazione con i lettori, sia per l'alimentazione del tag (nel caso di tag) passivi

I tag possono essere suddivisi in passivi ed attivi: i primi non contengono una batteria interna e vengono alimentati direttamente dal lettore, mentre i secondi contengono una batteria interna. Per tale motivo, le distanze di lettura risultano essere molto diverse tra tag passivi (da qualche cm a qualche metro) ed attivi (possono arrivare a centinaia di metri).

Il lettore

Il lettore è l'elemento che in un sistema RFID consente di ottenere i dati contenuti in un tag: il suo ruolo è pertanto quello di interprete tra il mondo fisico e quello applicativo. Il lettore contiene pertanto due interfacce: una di input (verso le antenne) e una di output (verso l'elaboratore, sia esso software o hardware). I componenti principali di un lettore sono:

- Un'unità di controllo, responsabile della comunicazione con l'applicativo
- Un'unità HF, formata da un trasmettitore ed un ricevitore, che gestisce il segnale da e verso il tag.
- Un'antenna per la trasmissione del segnale generato dall'unità HF e per l'alimentazione nel caso di tag passivi.

A.2.3 Frequenze

Una delle caratteristiche fondamentali di un sistema RFID è la frequenza operativa, ovvero la frequenza di trasmissione tra il lettore ed il tag. La comunicazione tra questi due elementi dipende sia dal tipo di tag e sia dall'applicazione utilizzata, e può differire da uno stato all'altro. Le frequenze possono essere divise come segue:

- **Basse frequenze (125-153 KHz):** i trasponders di questo tipo sono quasi sempre passivi e sono caratterizzati da una distanza di lettura bassa. I tag che lavorano a questa frequenza sono attualmente utilizzati nell'identificazione di animali.
- **Alte frequenze (10-15 MHz):** anche in questo caso i trasponders sono passivi e caratterizzati da una distanza di lettura medio-bassa. I tag che lavorano a questa frequenza sono attualmente i più utilizzati nell'identificazione di persone.
- **Altissime frequenze (850-950 MHz):** i tag che lavorano a questa frequenza sono attivi ed autoalimentati. Grazie all'altra distanza di lettura sono utilizzati nella logistica (tracciamento di pallet e containers).
- **Microonde (2.45-5.8 GHz):** i tag che lavorano a questa frequenza sono attivi ed autoalimentati. Sono utilizzati in sistemi che necessitano grosse distanze di lettura quali sistemi Telepass.

A.2.4 Standard

Le organizzazioni di standardizzazione nazionali ed internazionali stanno lavorando al fine di realizzare standard universalmente riconosciuti e risolvere le limitazioni che derivano dalle scelte differenti di ogni nazione. I due enti maggiori che stanno lavorando alla definizione di standard per la tecnologia RFID sono ISO ed EPCGlobal:

- ISO (International Organization for Standardization) ha definito una serie di standard già ampiamente utilizzati per l'identificazione di persone, animali e cose, tra cui lo standard ISO 14443 Proximity Cards che è stato utilizzato nel presente lavoro di tesi
- EPCGlobal (Electronic Product Code) ha definito diversi standard sia per l'identificazione dell'oggetto (tramite un codice univoco a livello mondiale), sia per le tecnologie di rete utili per interconnettere sistemi RFID diversi tra loro, al fine di creare un 'Internet delle cose'.

A.2.5 Applicazioni tipiche

Vi sono diversi scenari applicativi che sfruttano la tecnologia RFID per migliorare i processi. Alcuni di questi presentano dei sistemi che sono già ben sviluppati e testati; altre applicazioni per diversi motivi ancora sono presenti solo a livello di ricerca. Le applicazioni attualmente più sviluppate sono:

- Automazione della supply chain
- Tracciamento degli assets in uffici, laboratori, biblioteche.
- Applicazioni mediche quali cartelle cliniche informatizzate
- Tracciamento di persone e controllo accessi
- Tracciamento delle parti nell'industria manifatturieta
- Tracciamento di oggetti nei supermercati (scaffali intelligenti)
- Inventario in tempo reale degli oggetti in magazzino
- Identificazione di animali
- Tracciamento di veicoli in transito nelle stazioni autostradali
- Timing degli atleti durante gare sportive

A.2.6 Problematiche di sicurezza e privacy

L'identificazione a distanza di oggetti e/o persone ha portato al sollevamento di problematiche di privacy e di sicurezza non indifferenti. Il fatto che un tag possa contenere informazioni sensibili su una persona oppure che un pallet contenga informazioni critiche per il management di una società e che queste possano essere lette in modo del tutto trasparente, rende necessaria l'introduzione di tecniche di sicurezza al fine di evitare letture di tag a scopi malevoli. Diverse soluzioni sono state proposte, quali la disattivazione di tag dopo il loro utilizzo oppure l'adozione di portafogli in metallo che impediscono la lettura nel caso di documenti. Ma il fatto che ancora non vi siano degli standard riconosciuti universalmente comporta il non completo sviluppo di tali soluzioni. Sono comunque allo studio come per gli standard, delle soluzioni comuni anche per le questioni di sicurezza e di privacy.

A.3 L'idea

Dopo la breve panoramica sulle principali caratteristiche , componenti, standard della tecnologia RFID eseguita nel precedente capitolo, passiamo ora ad analizzare le potenzialità relative al suo utilizzo ed integrazione nei più comuni Sistemi Informativi aziendali. Negli ultimi 3 anni si è assistito ad una crescita rapida del settore legato alla tecnologia RFID. I continui investimenti e la definizione di standard di funzionamento hanno portato ad un notevole abbassamento dei relativi costi e rischi di utilizzo. Diverse applicazioni sono state sviluppate negli ultimi tre anni nei settori dell'automazione, trasporti e medicina. Solo adesso, dopo numerosi e rischiosi investimenti, il mercato sembra pronto ad accettare l'RFID come una tecnologia sulla quale investire. Tali presupposti spalancano le porte del mercato a quelle aziende intenzionate a sviluppare delle soluzioni software pronte a sfruttare al massimo le opportunità insite in tale tecnologia. Lo scopo di questo capitolo

è quello di analizzare le potenzialità della tecnologia RFID e lo stato attuale del mercato per delineare le caratteristiche principali di un Sistema Informativo utile a fornire un valido supporto alla gestione di strutture ricettive e/o commerciali. Al fine di ridurre la complessità d'analisi è necessario focalizzarsi su quelle strutture caratterizzate da processi/attività similari. Un'analisi delle relative caratteristiche porterà alla definizione di una serie di interventi di informatizzazione da comporre per la realizzazione di un completo sistema.

A.3.1 Analisi del contesto applicativo: resort, villaggi, poli fieristici, parchi

Il precedente studio sulle caratteristiche ed il mercato della tecnologia RFID ha spinto il nostro gruppo a prendere in considerazione l'opportunità di realizzare un sistema informativo utile alla gestione di strutture ricettive e/o commerciali. Numerose sono le aziende che trarrebbero elevati vantaggi da un relativo investimento. Tra tali tipologie di strutture, ideate per scopi di intrattenimento o relax, congressi e meeting, troviamo i resort turistici (hotel, lidi, villaggi), poli fieristici, parchi di divertimento e molto altro ancora. Tali attività sono tutte accomunate da una serie di elementi/procedure utili a garantire loro una corretta operatività. Un hotel, villaggio, lido, polo fieristico è solitamente caratterizzato da:

- Uno o più banchi di accettazione: rappresentano il punto di accesso alla struttura tramite cui viene registrato l'ingresso del cliente e conferito ad esso tutto il potenziale per usufruire dei servizi interni.
- Uno o più punti di accesso controllati: punti di controllo dei diritti di accesso del cliente. Tramite dispositivi automatici o personale interno il cliente dovrà dimostrare di possedere il diritto di accesso.
- Uno o più punti informativi: sono d'aiuto al cliente nel ricevere informazioni su luoghi interni, servizi e tutto ciò che si pone come obiettivo garantire ad esso un piacevole soggiorno.
- Uno più punti vendita: bar, boutique, completano l'offerta della struttura.
- Una o più attività eseguite all'interno: costituiscono punto di incontro e socializzazione per gli utenti. Ciascun cliente può iscriversi a tutte le attività desiderate tra quelle proposte dalla struttura. Un servizio di prenotazione tiene conto degli iscritti.

A.3.2 La nostra Idea

Considerando le principali caratteristiche delle strutture ricettive illustrate brevemente nel paragrafo precedente, la nostra idea è: realizzare un sistema capace di supportare e migliorare l'efficienza e l'efficacia dei processi caratteristici di tali strutture. Oltre a fornire un utile supporto strategico alla sfera manageriale, questo deve essere in grado di accompagnare e facilitare il cliente a godere tutto ciò cui è interessato, dal primo

accesso alla fruizione dei servizi, dall'acquisto di prodotti al bar alla partecipazione alle attività ricreative. Un aspetto molto importante è fare in modo che egli si senta parte di una comunità, i cui componenti sono accomunati dall'utilizzo di uno speciale accessorio integrante il tag RFID, che consente loro di liberarsi dall'utilizzo dei contanti e fruire di servizi veloci e personalizzati.

A.3.3 Analisi del contesto applicativo

Dopo aver identificato e descritto brevemente i principali elementi accomunanti le diverse tipologie di strutture ricettive, passiamo ad una attenta analisi di quest'ultimi, focalizzando la nostra attenzione su quegli aspetti che rappresentano le principali problematiche di operatività. L'identificazione di tali problematiche ci permetterà di definire una serie di interventi di informatizzazione, la cui composizione ci porterà alla realizzazione di un completo Sistema informativo basato sull'integrazione delle tecnologie RFID.

A.3.4 Procedura di check-in e controllo accesso

Tutte le attività utili alla registrazione dell'accesso del cliente sono svolte da un postazione localizzata all'ingresso della struttura. A valle del pagamento di una tariffa di ingresso, un abbonamento, l'affitto di una stanza o altro(differenti a seconda della tipologia di struttura), il cliente ne acquista il diritto di accesso. Una ricevuta, pass o altro, costituisce una prova dell'acquisita autorizzazione.

Problemi

Il conferimento dell'autorizzazione di accesso alla struttura al cliente, implica il necessario controllo della disponibilità delle risorse della stessa. Mentre la reception di un hotel o un villaggio turistico debbano accertarsi della disponibilità di stanze o appartamenti, il banco d'accettazione di un polo fieristico o di un parco di divertimenti, deve accertarsi che ci sia sufficiente spazio nell'area interna, per evitare il sovrappopolamento e conseguenti problemi di sicurezza. Attenti controlli sui diritti di accesso dei clienti devono essere operati, in modo da evitare che persone non autorizzate o malintenzionati possano fruire impropriamente dei servizi interni, o causare danni ai clienti. La risoluzione di tali problemi deve necessariamente passare:

- Dall'utilizzo di una procedura di gestione delle prenotazioni ed accessi
- Dall'utilizzo di oggetti dimostranti la veridicità dell'autorizzazione all'accesso

Opportunità di cambiamento

La vecchia e lenta procedure di gestione delle prenotazioni ed accessi tramite materiale cartaceo deve essere abolita per far posto ad una più veloce e precisa gestione tramite componenti informatici. L'utilizzo delle più moderne tecnologie informatiche (PC, database elettronici, software gestionali di prenotazione) consentirà di ottimizzare l'efficienza di tali task. I vecchi pass/badge(spesso cartacei) utilizzati per dimostrare ai predisposti punti di

controllo il diritto all'accesso/fruizione dei servizi dello stabilimento, sono risultati spesso poco affidabili in quanto facilmente clonabili e soggetti a facile usura. L'impiego di persone per il controllo in tali punti di accesso è risultato spesso poco efficace; parenti, amici, conoscenti vengono spesso ammessi senza averne il diritto. L'utilizzo di pass integranti tag RFID, impossibili da clonare, e di dispositivi automatizzati per il controllo dell'accesso (tornelli metallici) utili alla loro lettura tramite relativo lettore, consentiranno di controllare efficacemente gli accessi dei clienti. Nessun cliente che non sia stato registrato correttamente al banco d'accettazione potrà aver accesso alla struttura.

A.3.5 Desk informativi

Ideati per essere d'aiuto al cliente nel fruire dei servizi e beni e muoversi facilmente all'interno dello stabilimento, i desk informativi sono delle componenti essenziali per le strutture ricettive.

Problemi

La loro realizzazione comporta un non banale investimento in termini di risorse umane e denaro, da sottrarre ad una miriade di altre attività, che potrebbero risultare operativamente più importanti. Utilizzare delle risorse umane per svolgere task così statici (attendere che un cliente si presenti al desk, fornire ripetutamente le stesse informazioni) potrebbero indurre il management nell'erronea valutazione circa la scarsa utilità di tali punti. La realizzazione di un numero non sufficiente di punti informativi comporterebbe però, specie nei periodi di notevole affollamento, dei disagi a tutti coloro che desiderano avere velocemente delle informazioni.

Opportunità di cambiamento

La realizzazione di postazioni informatizzate per l'erogazione delle informazioni ai clienti, consentirebbe al management di ridurre consistentemente i costi, altrimenti onerosi, da sostenere per la realizzazione dei tradizionali desk informativi. Un chiosco/totem informatizzato, caratterizzato da un lettore RFID per restringerne l'utilizzo ai soli clienti, potrebbe consentire di visualizzare news e comunicati della direzione, ricaricare un eventuale conto virtuale associato al tag RFID, visualizzare il proprio profilo personale, prenotare/acquistare attività/servizi ed ottenere informazioni sulle aree della struttura.

A.3.6 Punti vendita di beni e servizi

Sono i punti di interesse incaricati della fornitura di cibo/bevande, vendita di servizi o sottoscrizioni ad attività ai clienti. Localizzati in punti strategici all'interno della struttura, sono in grado di soddisfare le principali necessità e desideri della clientela.

Problemi

Nelle strutture di medie e grandi dimensioni, la tradizionale gestione dei punti cassa mediante semplici registratori fiscali lascia eccessiva libertà agli impiegati incaricati dei pagamenti circa l'opportunità di sottrarre denaro alla società evitando la registrazione fiscale. Inoltre, in molte strutture come villaggi, lidi balneari o parchi acquatici, la clientela è solita utilizzare leggeri vestiti e/o costumi da bagno che non risultano spesso adatti ad ospitare accessori come portafogli o portamonete. Piuttosto che sopperire ad eventuali scomodità, molti di essi preferiscono lasciare i propri effetti in posti sicuri e più comodi, rinunciando spesso ad effettuare acquisti in tali momenti. Tale situazione comporta spesso un preoccupante decremento delle vendite alla clientela, sulle quali, invece, la maggior parte delle società basa il proprio profitto.

Opportunità di cambiamento

Date le sue caratteristiche, l'utilizzo di tag RFID indossabile, da usarsi come carta prepagata, consentirebbe di risolvere efficacemente le problematiche sollevate nel paragrafo precedente. Ogni punto vendita dovrebbe semplicemente essere dotato di un PC, un lettore RFID ed un modulo software deputato al controllo della validità delle transazioni. Ogni acquisto potrebbe essere in tal modo completato senza l'utilizzo di denaro contante annullando ogni tentativo di sottrazione del denaro da parte di malintenzionati, ed ogni transazione registrata nel database supportando strategicamente il management (analisi trend vendita).

A.3.7 Monitoraggio della struttura

Un bacino di utenza di centinaia o migliaia di persone implica un'attenta attività di monitoraggio mirante a minimizzare accessi non autorizzati, azioni illegali e danni a cose e/o persone. Usualmente equipaggiati con dispositivi elettronici come walkie talkie, auricolari, telefonini, gli impiegati incaricati delle attività di monitoraggio della struttura consentono al management di mantenerne il pieno controllo, evitando problemi di ogni sorta.

Problemi

Un controllo preciso dei diritti di accesso o di fruizione di specifici servizi da parte dei clienti, richiederebbe però, agli impiegati incaricati del monitoraggio, un aggiornamento real-time su ogni sorta di transazione eseguita (accessi, vendite ecc.) nella struttura. Con le tradizionali strumentazioni ciò risulta impossibile: qualora un impiegato abbia un dubbio sul diritto di accesso di un cliente, sarebbe costretto ad accompagnarlo al desk di accettazione per più approfonditi controlli, causando i relativi inconvenienti nel caso in cui sia caduto in errore.

Opportunità di cambiamento

Tra le funzionalità a disposizione del personale incaricato del controllo, vi dovrà essere quella di identificazione del cliente e di monitoraggio della struttura tramite palmare computerizzato(PDA). La sua connessione wireless ad un database, consentirà all'impiegato di avere informazioni costantemente aggiornate. Occasionalmente, egli potrà avviare una sorta di ‘ronda’ durante la quale dovrà disporre delle informazioni circa la situazione che viene supposta dal sistema: postazioni assegnate, servizi collegati e quant’altro. L’impiegato potrà così notare qualsiasi discrepanza, legata ad esempio alla presenza di clienti in una postazione che il sistema ritiene libera.

A.3.8 Il mercato: concorrenti

L’analisi completata nei paragrafi precedenti ci ha permesso di ipotizzare, analizzando le principali caratteristiche, interventi da operare sui principali processi ed attività caratterizzanti le strutture come poli fieristici, resort turistici, parchi. Dal momento che l’obiettivo di tale studio è quello di aggregare le nostre conoscenze accademiche per realizzare una concreta soluzione software, eventualmente commercializzabile, il nostro progetto non poteva prescindere da una, seppur breve, analisi dei principali competitor del settore di destinazione della nostra idea di soluzione. Dopo aver eseguito una breve analisi delle caratteristiche delle soluzioni offerte da due tra i competitor più importanti, abbiamo scoperto che ci potessero essere tutti i presupposti per realizzare una soluzione competitiva ed aggredire agevolmente un mercato che sembra ben lontano dall’essere inespugnabile.

A.4 Il case study

A.4.1 Introduzione

Le considerazioni mostrate nei capitoli precedenti permangono ad un livello di astrazione che rende impossibile procedere alla raccolta ed all’analisi dei requisiti. E’ necessario dunque focalizzarsi su un caso di studio per meglio comprendere problematiche ed opportunità afferenti al dominio di applicazione. La Mondello Immobiliare Italo Belga SA (per brevità MIIB) ha in concessione 2 chilometri di spiaggia nel comune di Palermo. Con la sua collaborazione è stato possibile focalizzarsi su un chiaro caso di studio: una spiaggia attrezzata.

A.4.2 Applicabilità ad una spiaggia attrezzata

Richiamiamo le caratteristiche generali della struttura tipo presentata al capitolo 3, inquadrandole stesse nel contesto di una spiaggia attrezzata, così da capire l’applicabilità delle nostre considerazioni al caso specifico:

- uno o più botteghini d’ingresso: chiunque voglia accedere alla spiaggia attrezzata deve pagare un biglietto d’ingresso al botteghino.

- uno o più punti di accesso controllati: la maggior parte delle spiagge attrezzate presenta una procedura di controllo degli accessi piuttosto flessibile. I clienti devono essere controllati al loro arrivo ma devono anche essere abilitati ad uscire e rientrare in qualsiasi momento; precise disposizioni legislative impediscono inoltre un controllo totale degli accessi alla struttura (basti pensare all'accesso dal mare). Il controllo degli accessi è dunque assai ostico e non affidabile ad un botteghino già oberato di lavoro.
- uno o più banchi informativi: come qualsiasi struttura che ospita centinaia di persone ogni giorno, è assolutamente necessario fornire informazioni alla clientela.
- uno o più punti vendita: un bar all'interno della spiaggia costituisce già un punto vendita che sarebbe interessante automatizzare.
- delle attività svolte all'interno: diverse le attività che si possono organizzare in una spiaggia attrezzata al fine di rendere più piacevole la permanenza ai clienti.

A.4.3 La nostra reingegnerizzazione dei processi

Abbiamo appena visto come una spiaggia attrezzata costituisce un perfetto contesto applicativo per un SI che sfrutti la tecnologia RFID. Tuttavia, nessuna compagnia intraprenderebbe mai un progetto del genere senza un'attenta analisi di costi e benefici. L'approccio più consono alle nostre capacità è costituito dalla predisposizione di uno studio di fattibilità che sia in grado di fornire al direttivo della MIIB l'insieme di informazioni necessarie all'avviamento reale del progetto. Queste informazioni riguardano la fattibilità tecnica ed organizzativa, i benefici, i costi, i rischi e le scadenze previste per la realizzazione del SI in esame che chiameremo: Mondello IT Be@ch. L'analisi della gestione operativa delle spiagge attrezzate gestite dalla MIIB ha evidenziato una serie di problematiche ed opportunità. La tabella 4.1 è divisa nelle seguenti colonne:

1. Problemi esistenti rilevati;
2. Opportunità di tipo generale: opportunità che prescindono dall'introduzione del sistema in esame;
3. Opportunità emergenti nell'ambito del progetto: opportunità che emergono nel prendere in considerazione l'intervento di strumenti informatici a supporto della gestione.

Compatibilmente con il processo di revisione di diversi processi aziendali e di miglioramento dell'immagine societaria sono stati identificati gli obiettivi di progetto riportati nella tabella 4.2.

La società Mondello intende adottare un sistema informativo(SI) che:

- fornisca una soluzione ad un certo numero di problematiche di gestione delle porzioni di litorale adibite a spiaggia attrezzata;

A – Riassunto in lingua italiana

Problemi esistenti rilevati	Opportunità di tipo generale	Opportunità emergenti nell'ambito del progetto
Mancanza di un controllo efficace all'ingresso della spiaggia attrezzata (favoritismi)	Monitoraggio degli ingressi per la raccolta di informazioni sull'andamento delle presenze nella struttura	Gestione informatizzata degli accessi tramite tornelli dotati di lettore RFID Raccolta di informazioni dettagliate e computazione di statistiche sui trend delle presenze
Mancanza di un controllo efficace nella gestione del servizio di affitto di arredo balneare all'interno della spiaggia attrezzata	Liberare i clienti dall'uso di denaro contante all'interno della struttura	Monitoraggio della fruizione dell'arredo balneare attraverso un'assegnazione informatizzata delle postazioni
	Monitoraggio della vendita di servizi e beni offerti all'interno della struttura finalizzato all'attivazione di un programma di fidelizzazione dei clienti	Dotazione dei clienti di un sistema di pagamento elettronico basato su uno strumento "indossabile". Raccolta di informazioni dettagliate e computazione di statistiche sui trend d'acquisto dei clienti
	Vendita di spazi pubblicitari all'interno della struttura	Realizzazione di un sistema di rotazione intelligente di pubblicità su schermo piatto vincolata al tipo di clientela presente nella struttura istante per istante

Figura A.1. Problemi e opportunità'

Obiettivi di progetto
Controllo fine sugli accessi
Controllo fine sull'attrezzatura
Rendere obsoleto l'uso dei contanti
Dare supporto alle decisioni manageriali
Minimizzare l'impatto su personale e clienti
Migliorare l'immagine della società Mondello
Rispetto dei vincoli economici e temporali

Figura A.2. Obiettivi di progetto

- consenta al contempo di sfruttare anche alcune opportunità latenti;
- costituisca un ulteriore impulso verso il rilancio dell'immagine societaria, in linea con le scelte strategiche adottate finora dal direttivo corrente.

Il progetto consiste dunque nella realizzazione di un SI che automatizzi i processi di cui si compone la gestione di una spiaggia attrezzata e che al contempo fornisca un valido supporto alle decisioni manageriali di breve, medio e lungo termine. Il progetto deve prendere il via da una situazione in cui non è presente alcun SI preposto all'assolvimento anche parziale delle funzioni che saranno interessate dal sistema in esame. Lo sviluppo consisterà di una parte dedicata alle componenti software e di una dedicata all'integrazione di tale software con una serie di componenti hardware cosiddetti off-the-shelf (acquistati da terzi e non realizzati internamente). La società Mondello ha mostrato il proprio interesse

nel raggiungimento di una serie di risultati specifici. Ad essi sono riconducibili altrettanti obiettivi di progetto volti al loro raggiungimento, come mostrato dalla tabella 4.3.

Risultati	Obiettivi
Miglioramento dell'immagine societaria	Realizzazione di un sistema basato su una tecnologia innovativa e di tendenza (RFID) che incontri il favore di tutti gli stakeholder positivi di progetto.
Incremento della qualità dei servizi resi al cliente	Realizzazione di un sistema che incrementi il livello di efficienza, efficacia e soddisfazione dell'utente legato ai servizi erogati
Controllo sugli accessi automatico ed efficace che precluda l'accesso ai non paganti	Implementazione di un meccanismo automatizzato che vincoli in modo imprescindibile l'accesso alla struttura al pagamento del biglietto previsto
Diffusione dell'uso di un sistema di pagamento elettronico proprietario all'interno della struttura balneare	Realizzazione di un sistema che mostri ai clienti chiari vantaggi legati alla praticità dell'uso di un sistema di pagamento elettronico basato su tag RFID
Ottenimento di una solida base di informazioni su cui fondare scelte manageriali di breve, medio e lungo termine	Realizzazione di un sistema in grado di tenere traccia di tutti gli eventi riconducibili a trend di frequenza alla struttura e di acquisto, con lo scopo di rielaborare i dati immagazzinati fornendone una vista riepilogativa ed indicativa
Controllo sulla liceità della fornitura di arredo balneare e servizi ai clienti all'interno della struttura	Implementazione di un meccanismo di "ronde" periodiche -coadiuvate da strumenti informatici- volte alla determinazione di eventuali irregolarità legate alla fornitura di arredo balneare e servizi

Figura A.3. Risultati e obiettivi

Per facilitare l'identificazione dell'ambito di progetto è consigliabile inquadrarlo in termini di prodotti finali che il progetto intende realizzare. Il progetto dunque include:

- Realizzazione del SI Mondello IT Be@ch conforme ai requisiti che saranno formalizzati nel RAD
- Installazione del sistema nella location desiderata
- Training del personale
- Manuale utente a supporto del SI
- Codice sorgente del SI
- Documentazione di progetto
- Documentazione del SI
- Assistenza

Ne rimangono invece esclusi:

- Piano di pubblicizzazione dell'intervento di informatizzazione
- Allocazione (vendita) degli spazi pubblicitari previsti dal sistema
- Piano di marketing volto ad uno sfruttamento ottimale delle potenzialità del sistema
- Interventi di modifica non programmati alle funzionalità del SI dopo la sua installazione in loco
- Aggiornamenti del software ed Upgrade annuali

Tra i modelli di ciclo di vita del progetto presi in considerazione (Cascata, Incrementale, Evoluzionario), il modello più adatto al nostro progetto è l'Evoluzionario, caratterizzato da un alto livello di flessibilità nei requisiti in progetti di media dimensione, soprattutto quando il dominio di applicazione non è conosciuto a priori.

A.4.4 Identificazione dei processi e raccolta dei requisiti

La prima sezione del Requirements Analysis Document contiene la nostra disamina della situazione della MIIB prima dell'introduzione del SI, portata avanti attraverso l'ispezione dei processi, delle procedure interne e delle politiche impattate dall'introduzione di un sistema come quello in esame. La MIIB riserva una parte del litorale ad un certo numero di capanne concesse in affitto per l'intera stagione. Accanto a questo storico modello di business, la MIIB ha deciso da poco per la destinazione di porzioni sempre più grandi di spiaggia al modello di spiaggia attrezzata. Nonostante la MIIB non fosse del tutto preparata a questo cambio di rotta, la clientela sta fornendo dei riscontri assai positivi, e la MIIB è dunque convinta a procedere. Ciò che sta comunque succedendo, è che la MIIB sta testando il proprio modello manageriale sul campo e non può che essere fondamentale uno strumento che sia in grado di raccogliere il maggior numero possibile di indicatori a supporto di un'analisi dei trend di gestione.

A.5 Analisi e design del sistema

Come abbiamo visto nel capitolo precedente, il sistema individuato dovrà supportare un'ampia gamma di processi e abilitare un'esecuzione strutturata delle numerose attività che gravitano attorno alla gestione della spiaggia attrezzata. Prima dell'introduzione del sistema tali processi erano totalmente basati su documenti cartacei, pertanto per il design della soluzione nessuna problematica di compatibilità è stata affrontata. Lo scopo di questo capitolo è quello di affrontare la parte di analisi dei requisiti, al fine di scoprire le funzionalità di ogni modulo, e la parte di design al fine di capire come abbiamo deciso di sviluppare i vari moduli. Iniziamo con l'analisi dei requisiti.

A.5.1 L’analisi dei requisiti

Approccio all’analisi dei requisiti

I requisiti ottenuti attraverso il processo di analisi ed interviste con gli impiegati ed i managers sono spesso in un formato grezzo. Al fine di raggiungere un modello formale utilizzabile per la scrittura di applicazioni bisogna raffinare questi dati in un set di modelli. Uno degli approcci più utilizzati è la modellazione tramite diagrammi e pattern UML¹, e infatti questo è stato il modello scelto per lo sviluppo del nostro progetto. Sono stati prodotti in ordine i seguenti modelli: Scenari, Casi d’uso, Diagrammi dei casi d’uso, Diagrammi delle classi, Diagrammi di sequenza, Diagrammi di stato, Diagrammi di attività.

Principali funzionalità

Elenchiamo adesso i diversi moduli individuati e le funzionalità principali che sono state identificate in tali moduli grazie ai diagrammi UML sopra elencati.

Modulo Botteghino Il modulo botteghino è tra tutti quello che presenta il maggior numero di funzionalità. Queste possono essere divise in 2 blocchi:

- Funzionalità per il check-in dei clienti
- Funzionalità per la gestione dei clienti

La funzionalità di check-in a sua volta può essere divisa in 2 tipi: check-in per l’accesso giornaliero e nuovo abbonamento. La prima funzionalità deve essere resa più celere possibile a causa delle code che potrebbero essere create dal momento che i clienti che vogliono effettuare l’accesso giornaliero si recano direttamente in spiaggia per effettuare l’accesso. La seconda funzionalità è più complessa e richiede più passaggi, ma risulta essere meno critica dal punto di vista delle tempistiche. Un’ulteriore funzionalità presente in questa sezione è inoltre l’apertura manuale dei gate di ingresso nel caso in cui la lettura automatica presenti dei problemi. La funzionalità di gestione dei clienti permette di visualizzare, ed eventualmente modificare, le informazioni riguardanti i clienti che hanno effettuato un accesso giornaliero o che hanno sottoscritto un abbonamento. L’accesso ad un cliente specifico può avvenire tramite la lettura del tag ad egli associato oppure tramite l’immissione di parametri. Le funzionalità specifiche per il cliente sono: modifica dati personali, modifica password, visualizzazione servizi ed attività acquistati, stampa della ricevuta per usufruire di un servizio o di un’attività, check-out di un cliente giornaliero (o terminazione di una sottoscrizione), visualizzazione delle transazioni, sostituzione del tag, sostituzione della locazione, ricarica del conto virtuale, aggiuzione ospite a sottoscrizione periodica.

¹Unified Modeling Language

Modulo interfaccia remota Il modulo interfaccia remota è quello che in un sistema software è comunemente chiamato ‘Back-office’ o ‘Back-end’. La sua funzionalità principale è quella di riempire il database con informazioni che saranno necessarie a tutti gli altri moduli per poter funzionare correttamente. In altre parole, è utilizzato dall’amministrazione per controllare tutte le tabelle statiche del database. Per questa ragione, le funzionalità di questo modulo sono definite come CRUD (dall’inglese Creare, Read, Update, Modify - Crea, Leggi, Aggiorna, Modifica). Elenchiamo di seguito i tipi di dato che in questo modulo vengono gestite con le funzionalità CRUD: attività, spiagge, categorie di cliente, sconti, impiegati, mappe locazioni, tipi di locazione, news, preferenze, tag, tipi di pagamento, tipi di sottoscrizione.

Modulo BAR Il modulo di pagamento al bar è la parte del sistema che permette di effettuare il pagamento al bar tramite il tag RFID collegato al conto virtuale del cliente. L’obiettivo principale è quinti l’utilizzo del tag come una carta di debito. Questo modulo funziona insieme al software gestionale RIGEL, che risulta essere già installato presso il punto bar della struttura. La funzionalità di questo modulo è pertanto quella di controllare se il cliente che vuole effettuare il pagamento possiede abbastanza credito, ed in caso positivo aggiornare il database e comunicare al gestionale del bar il corretto pagamento.

Modulo gestione tornelli Il modulo gestione tornelli è la parte del sistema che gestisce l’accesso alla struttura tramite tornelli posti all’ingresso. I tornelli contengono un lettore RFID, e pertanto consentono al cliente di utilizzare il tag RFID come un pass d’ingresso. La funzionalità è pertanto quella di controllare i diritti d’accesso del cliente che esibisce il tag e di decidere se questi ha il diritto d’accesso alla struttura.

Modulo chiosco informatizzato Il modulo chiosco informatizzato viene installato nei totem sparsi per la struttura, i quali sono composti da un touch screen, da un lettore rfid, da un lettore di banconote e da una stampante termica. Tali device sono liberamente utilizzabili dai clienti senza l’ausilio di alcun impiegato: sono infatti pensati per rendere la gestione servizi più veloce evitando inutili code al botteghino. Le funzionalità offerte sono:

- Lettura notizie e annunci pubblicati dall’azienda
- Acquisto servizi
- Sottoscrizione ad attività
- Visualizzare i servizi ed attività acquistati e stampare le ricevute
- Ricaricare il proprio conto virtuale
- Visualizzare le transazioni effettuate tramite il pagamento RFID
- Visualizzare e modificare i dati personali
- Modificare la password personale

A.5.2 Design del sistema

L'approccio di design scelto per il presente progetto è l'*iterativo incrementale*: la prima architettura che viene prodotta tramite tale approccio è quella di alto livello, che può essere testata mediante scenari chiave, requisiti, vincoli conosciuti, attributi di qualità ed il frame dell'architettura. Dopo aver creato l'architettura candidata ad essere quella definitiva, sebbene ancora non completa, è possibile acquisire maggiori informazioni circa il design ed è possibile avere degli scenari chiave più dettagliati: tali informazioni permettono di reiterare il processo di design e pertanto l'architettura creata può essere rifinita fino a quando non si raggiunge il livello di dettaglio desiderato. Il sistema da implementare fonda il suo funzionamento sulle informazioni contenute in un database accessibili in modo differente da una serie di sottosistemi indipendenti fra loro. Il paradigma di funzionamento non prevede dunque l'instaurazione di un mezzo di comunicazione fra sottosistemi di per sé inutile: questo paradigma ha portato alla selezione del modello cosiddetto a Repository. Esso prevede la presenza di un database centrale con il quale i sottosistemi interagiscono secondo il paradigma inserisci-modifica-cancella. Nonostante una grossa percentuale delle informazioni di supporto al sistema siano condivise tra più sottosistemi, esse dovranno essere fruite on demand (solo quando richieste dai sottosistemi). Sarà dunque possibile limitarsi ad implementare un adeguato sistema di controllo dell'accesso concorrente ai dati che escluda la possibilità di una violazione dell'integrità dei dati. Tra ogni sottosistema ed il database dovrà essere presente dunque un sistema di gestione noto come DBMS, utile a gestire l'accesso ai dati e la concorrenza tra diversi sottosistemi. Di seguito viene mostrata una figura che aiuta a comprendere l'adesione al modello a Repository.

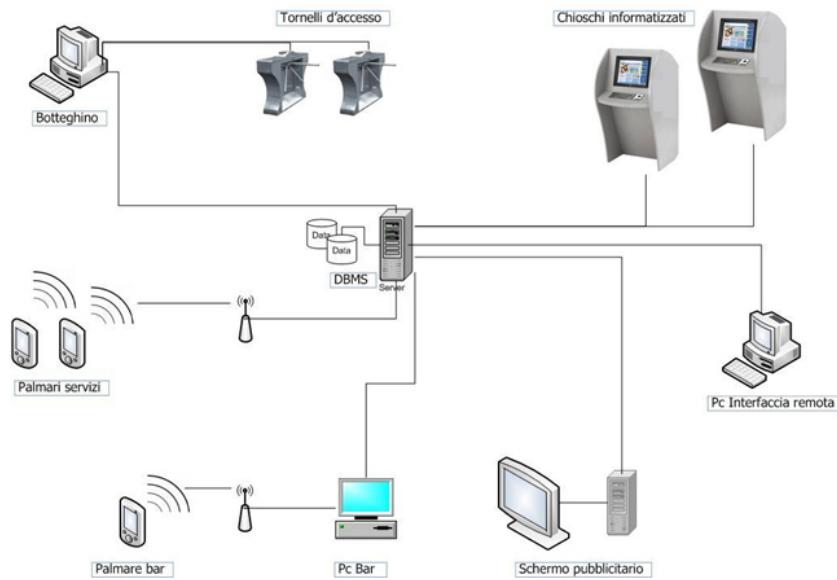


Figura A.4. Il modello a repository per il nostro progetto

I sottosistemi individuati sono gli stessi elencati nel paragrafo precedente (analisi dei

requisiti). Vediamo in breve le scelte di design applicate per tali moduli:

- **Modulo Interfaccia remota:** per questo modulo è stata scelta un'applicatione di tipo 'Rich Client Application' ed uno stile architetturale *layered* con 3 livelli.
- **Modulo botteghino:** per questo modulo è stata scelta un'applicatione di tipo 'Rich Client Application' ed uno stile architetturale *layered* con 3 livelli.
- **Modulo Chiosco informatizzato:** per questo modulo è stata scelta un'applicatione di tipo 'Rich Client Web Application' ed uno stile architetturale *layered* con 3 livelli.
- **Modulo Gestione Tornelli:** per questo modulo è stata scelta un'applicatione di tipo 'Service Application' ed uno stile architetturale *layered* con 2 livelli.
- **Modulo Pagamento BAR:** per questo modulo è stata scelta un'applicatione di tipo 'Service Application' ed uno stile architetturale *layered* con 2 livelli.

A.6 Sviluppo del sistema

A.6.1 Layering

Sulla base dell'esperienza del team di progettazione, e del risultato della fase di design, lo stile architetturale prescelto per l'implementazione di tutti i sottosistemi componenti IT-Beach è risultato essere quello denominato Layered o stratificato. Tale stile architetturale si basa sulla scomposizione delle componenti del sottosistema nei seguenti livelli logici:

- Presentation, livello logico contenente componenti dedicate al rendering delle interfacce grafiche e gestione delle interazioni con l'utente.
- Business logic, livello logico contenente componenti dedicate alla gestione della logica dell'applicativo in termini di elaborazione dati, esecuzione di algoritmi, gestione delle interfacce grafiche (avvio, chiusura).
- Data Access, livello logico contenente componenti dedicate alla gestione delle entità dati e delle procedure di accesso a DB locali o remoti.

La struttura a Layer garantisce inoltre elevata Manutenibilità ed Estendibilità grazie alla scomposizione delle componenti del sistema in aree di competenza differenti. Il sistema risulta pertanto facilmente modificabile e controllabile nel caso in cui sia necessario apportarvi delle modifiche in un futuro. Il Layered consente pertanto di realizzare delle architetture snelle, performanti e semplici da gestire nel tempo, consentendo di fatto di raggiungere molti degli obiettivi proposti durante la fase di design. Numerosi sono gli strumenti e tecniche di implementazione proposte dalla letteratura per la realizzazione delle singole componenti e la struttura di ciascun livello. Guidati dagli obiettivi di design e dalle community di sviluppatori abbiamo scelto le più appropriate alla nostra soluzione.

A.6.2 Considerazioni sul Data Layer dell'intero sistema

Per la realizzazione delle componenti del Data Layer di tutti i moduli è stata scelto lo strumento per la generazione di codice automatico . netTiers. Tecnologia open source basato su modelli di generazione CodeSmith, semplifica i compiti di creazione del livello Data Layer per applicazioni personalizzate sviluppate in ambiente .Net. Utilizzando i metadati di database SQL Server, è in grado di creare tutte le componenti DALC(Data Acces Logic Components) per eseguire in maniera performante e sicura le più comuni operazioni di lettura, modifica, ricerca, cancellazione alle tabelle del database. Ciascun modulo potrà pertanto utilizzare i metodi e gli oggetti che più interessano per l'implementazione delle proprie funzionalità.

A.6.3 Decomposizione del sistema

Modulo Botteghino

Il modulo Box Office è stato realizzato per automatizzare le procedure di check-in dei clienti alla struttura. Installato al desk di accettazione, consentirà ai clienti di sottoscrivere uno o più abbonamenti temporali(giornaliero,mensile, bimestrale ecc) per usufruire dei servizi e beni offerti all'interno. Dopo l'inserimento dei principali dati sulla tipologia dell'abbonamento, la visualizzazione della relativa mappa, consentirà al cliente di scegliere tra le postazioni disponibili quella che più gli aggrada. Confermata la postazione, inseriti i dati di uno o più clienti che la occuperanno, e consegnati i bracciali RFID, l'impiegato chiuderà la registrazione abilitando il/i cliente/i all'accesso.

Funzionalità All'avvio dell'applicativo, il sistema carica i dati statici(tipologie di abbonamento, sconti, mappa, categorie utenti ecc.) in un apposita cache pensata per migliorare le performance, verifica la presenza del lettore RFID abilitante l'assegnazione dei tag ai clienti e richiede il login dell'impiegato che, immesse le sue credenziali, può così accedere alle seguenti funzionalità:

- Nuova sottoscrizione: basata su una procedura a Wizard(a step), guida l'utente nella creazione di una nuova sottoscrizione periodica(mensile, bimestrale ecc.) in maniera semplice e veloce.
- Nuovo Accesso giornaliero: simile alla procedura precedente(condividono alcuni step), consente di abilitare il cliente ad accedere alla struttura per tutto l'arco della giornata.
- Gestione clienti: racchiude tutte le funzionalità legate alla ricerca e modifica dei parametri dei clienti (tag assegnati, credenziali ed info generiche) e ricerca, modifica e cancellazione dei parametri delle sottoscrizioni(cambio postazione, aggiunta ospiti, chiusura sottoscrizione, ecc.).
- Chiusura giornata: Al termine della giornata sarà possibile stampare un resoconto di tutti gli incassi suddivisi per tipologie di transazioni.

Scelte chiave di sviluppo Esaminiamo brevemente le scelte di sviluppo effettuate per ciascuno dei livelli della struttura Layered enfatizzandone le componenti chiave. Ricordiamo che per l'implementazione del Data Layer si è deciso di utilizzare lo strumento .netTiers, brevemente introdotto nel paragrafo relativo alle considerazioni su Data Layer.

Presentation layer L'interfaccia è basata su una struttura chiamata interfaccia multi-dокументo (MDI) , una metodologia di creazione di interfacce utente per applicazioni che consentono all'utente di lavorare con più di un documento simultaneamente. Il modulo utilizza il componente denominato ParentForm come finestra principale dell'applicazione. Le finestre generate per l'esecuzione delle sue funzionalità sono denominate finestre figlie e visualizzate sempre all'interno dei limiti della Parentform. Le componenti grafiche sono inoltre separate dai componenti che realizzano la logica dell'interazione secondo il paradigma previsto dalla libreria User Interface Process Application Block(UIPAB) della Microsoft Corporation.

Business Layer Il sottosistema basa le sue funzionalità sull'utilizzo di componenti business, altresì chiamati Controllori, che offrono servizi utili alla gestione del cuore logico dell'applicativo. Operazioni su database(lettura, scrittura), esecuzione di complessi algoritmi, elaborazione di dati, interfacciamento con hardware specifico(lettore RFID) sono solo alcuni dei complessi task alla base delle funzionalità offerte all'utente.

Piattaforma e dispositivi Il modulo è stato sviluppato in linguaggio C#, sfruttando la tecnologia Windows Forms offerta dalla strumento Microsoft Visual Studio basato sulla piattaforma .Net. La sua installazione richiede l'utilizzo di un PC o laptop di fascia media, un lettore RFID, un stampante termica di ricevute da 80 mm ed una connessione alla rete locale cui è connesso il server contenente database reso disponibile da SQL SERVER 2005 EXPRESS EDITION.

Modulo Interfaccia Remota

Modulo necessario alla gestione(inserimento, modifica, cancellazione) dei dati di configurazione necessari alla corretta operatività dei moduli del sistema. Le informazioni relative a:

- Tag, Categorie clienti, Locazioni, Opinioni, Attività, Servizi, News, Impiegati, Sconti, Mappa locazioni

e quant'altro funga da base alle funzionalità dei moduli Botteghino, Chiosco, Pagamento Bar, Tornelli potranno essere comodamente gestite dall'amministratore del sistema sia in una fase precedente che successiva alla prima installazione.

Funzionalità All'avvio dell'applicativo, il sistema carica i dati statici in un apposita cache pensata per migliorare le performance, verifica la presenza del lettore RFID abilitante la gestione delle informazioni sui tag e richiede il login dell'impiegato che, immesse le sue credenziali, può così accedere alle funzionalità predisposte alla gestione dati

per tutte le tabelle indicate nel precedente paragrafo e contenenti le informazioni semi-statiche(informazioni inserite massicciamente in fase di configurazione, raramente sottoposte a modifiche, sono utilizzate per essere composte e realizzare le funzionalità dei moduli) di ITBe@ch.

Scelte chiave di sviluppo Esaminiamo brevemente le scelte di sviluppo effettuate per ciascuno dei livelli della struttura Layered enfatizzandone le componenti chiave. Ricordiamo che per l'implementazione del Data Layer si è deciso di utilizzare lo strumento .netTiers, brevemente introdotto nel paragrafo relativo alle considerazioni su Data Layer.

Presentation layer L'applicazione richiede la realizzazione di interfacce per l'inserimento, modifica e cancellazione dei dati delle tabelle di configurazione. Come già avvenuto per il modulo Botteghino, anche il livello Presentation di Interfaccia Remota è basato sulla tecnologia MDI. Particolare attenzione è stata inoltre posta sulle procedure di validazione dei dati inseriti dall'utente.

Business Layer Il sottosistema basa le sue funzionalità sull'utilizzo di componenti business, altresì chiamati Controllori, che offrono servizi utili alla gestione del cuore logico dell'applicativo. L'implementazione dei servizi offerti da quest'ultimi non hanno richiesto un eccessivo sforzo data la grande semplicità delle operazioni da eseguire(solo CRUD). Particolare attenzione è stata invece posta sulle procedure di validazione dei dati inseriti dall'utente.

Piattaforma e dispositivi Il modulo è stato sviluppato in linguaggio C#, sfruttando la tecnologia Windows Forms offerta dalla strumento Microsoft Visual Studio basato sulla piattaforma .Net. La sua installazione richiede l'utilizzo di un PC o laptop di fascia media, un lettore RFID ed una connessione alla rete locale cui è connesso il server contenente database reso disponibile da SQL SERVER 2005 EXPRESS EDITION.

A.6.4 Modulo di pagamento al Bar

Il modulo dedicato al pagamento presso il punto di vendita del Bar consente di realizzare il paradigma di pagamento mediante tag RFID. Il cliente potrà infatti utilizzare il proprio bracciale RFID non solo per aver accesso ai tornelli, o al chiosco ma anche per effettuare i propri acquisti al bancone del Bar. Per la gestione delle ordinazioni e delle sale del locale Bar, è stato previsto l'acquisto ed utilizzo di un gestionale denominato RIGEL, prodotto dalla APS s.n.c. Il modulo bar dovrà interagire con tale software per integrarvi tale funzionalità di pagamento. L'integrazione ha richiesto una stretta e continua collaborazione con gli sviluppatori e la dirigenza della APS al fine di delineare la miglior strategia di comunicazione ed integrazione dei due moduli.

Funzionalità Data la necessità di offrire una singola funzionalità al software RIGEL, il modulo pagamento Bar è stato implementato come un servizio automaticamente avviabile

al bootstrap del sistema operativo. Il nostro modulo deve infatti essere sempre attivo e capace di operare con RIGEL al momento della chiusura e pagamento del conto cliente. Ogni qualvolta l'impiegato deve procedere al pagamento del conto del cliente tramite bracciale RFID, egli:

- Esegue la chiusura del tavolo del cliente tramite le funzionalità di RIGEL, calcolando il totale e scegliendo il metodo di pagamento RFID.
- Chiede all'utente l'inserimento della password utile all'autenticazione del pagamento.
- Confermato l'inserimento del PIN, attiva il modulo Bar che abilitante il lettore RFID tramite cui leggere adesso il bracciale del cliente.
- Dopo aver posto il bracciale campo operativo dell'apposito lettore, il sistema legge il seriale del Tag, prelevando i dati relativi al conto del cliente ed restituendo il risultato della transazione d'acquisto(conferma, cancellazione).

Scelte chiave di sviluppo Esaminiamo brevemente le scelte di sviluppo effettuate per ciascuno dei livelli della struttura Layered enfatizzandone le componenti chiave. Ricordiamo che per l'implementazione del Data Layer si è deciso di utilizzare lo strumento .netTiers, brevemente introdotto nel paragrafo relativo alle considerazioni su Data Layer. Non richiedendo una interazione con l'utente, in quanto deve solo comunicare con il software RIGEL, si è deciso di sviluppare l'applicazione come servizio e quindi di non realizzare alcun Presentation Layer.

Business Layer Ciascuna transazione di pagamento tramite tag RFID, è basata sullo scambio di 2 file testuali, richiesta e risposta di pagamento, contenenti i dati utili al completamento della transazione d'acquisto(PIN utente, codice transazione RIGEL, totale da pagare il primo ed esito pagamento, messaggio d'esito il secondo). La logica di base si articola nei seguenti passi:

- Lettura del file di richiesta e del tag RFID mediante libreria di comunicazione sviluppata dal nostro team
- Reperimento dei dati e controlli sui dati necessari a concludere il pagamento (validità PIN, tag assegnato, credito sufficiente, transazione registrata sul DB).
- Creazione del file di risposta per la comunicazione dell'esito della transazione.

Piattaforma e dispositivi Il modulo è stato sviluppato in linguaggio C#, sfruttando la tecnologia Windows Service offerta dalla strumento Microsoft Visual Studio basato sulla piattaforma .Net. La sua installazione richiede l'utilizzo di un PC o laptop di fascia media, un lettore RFID ed una connessione alla rete locale cui è connesso il server contenente il database reso disponibile da SQL SERVER 2005 EXPRESS EDITION. Nella scelta delle componenti HW sono stati tenuti in considerazione anche i requisiti minimi di funzionamento di RIGEL.

A.6.5 Chiosco informativo

Il sottosistema chiosco informatizzato è una soluzione studiata su misura del cliente per l'erogazione di servizi e beni offerti dalla struttura. Mira a ridurre il carico lavorativo sostenuto dal Botteghino(già elevato a causa delle procedure di abbonamento) offrendo al cliente una serie di funzionalità utili.

Funzionalità Il cliente potrà accedere alla schermata principale solo dopo aver effettuato l'accesso tramite lettura del proprio tag RFID. Alla comparsa di tale schermata, egli potrà scegliere di:

- Caricare il proprio conto legato al tag RFID assegnato, inserendo le banconote nel dispositivo di accettazione di banconote preposto
- Gestire il proprio profilo personale
- Modificare il pin per il pagamento sicuro
- Visualizzare la situazione del conto ed i movimenti eseguiti(ricarica, pagamenti al bar)

L'accesso alle aree dedicate alla visualizzazione della situazione contabile e del profilo prevedono l'autenticazione del cliente con inserimento dello stesso PIN usato per i pagamenti.

Scelte chiave di sviluppo

Esaminiamo brevemente le scelte di sviluppo effettuate per ciascuno dei livelli della struttura Layered enfatizzandone le componenti chiave. Ricordiamo che per l'implementazione del Data Layer si è deciso di utilizzare lo strumento .netTiers, brevemente introdotto nel paragrafo relativo alle considerazioni su Data Layer.

Presentation layer Lo sviluppo di un'applicazione a stretto contatto con il cliente, ha richiesto una notevole attenzione su quegli aspetti che dovevano garantire elevata affidabilità e facilità di interazione. Attente validazioni degli input dell'utente(validazione dati inseriti, pressione tasti speciali) ed attenti accorgimenti sulla definizione delle interfacce grafiche(pulsanti più grandi, testo esposto in modo chiaro e conciso) hanno permesso di realizzare un sistema affidabile e veloce.

Business Layer Il sottosistema basa le sue funzionalità sull'utilizzo di componenti business, altresì chiamati Controllori, che offrono servizi utili alla gestione del cuore logico dell'applicativo. Operazioni su database(lettura, scrittura), elaborazione di dati, interfacciamento con l'accettatore di banconote, sono solo alcuni dei complessi task alla base delle funzionalità offerte all'utente.

Piattaforma e dispositivi Il modulo è stato sviluppato in linguaggio C#, sfruttando la tecnologia Windows Forms offerta dalla strumento Microsoft Visual Studio basato sulla piattaforma .Net. La sua installazione ha richiesto la progettazione di un vero totem informativo in acciaio zincato per il quale sono stati applicati gli accorgimenti necessari(saldature, ventilazione ecc.) a garantirne una corretta operatività in ambienti altamente umidi e salini come lo sono quelli situati vicino al mare. Una connessione alla rete locale gli permette l'accesso al DB server reso disponibile da SQL SERVER 2005 EXPRESS EDITION.

A.6.6 Modulo di gestione tornelli

È un modulo utilizzato per eseguire la procedura di controllo degli accessi dei clienti alla struttura. Basato sull'impiego di tornelli integranti lettori RFID, è in grado di eseguire veloci controlli sulla validità di accesso della clientela negando o abilitandone il passaggio.

Funzionalità

Installato come Servizio Windows sul Database Server, si pone continuamente in ascolto di messaggi di richiesta d' accesso inviati dai tornelli connessi alla stessa rete locale. Il dispositivo integrato nei tornelli esegue infatti una continua lettura di tutti i tag RFID che entrano nel proprio campo operativo, creando, per ciascun di essi, un pacchetto UDP contenente il codice UID del tag da inviare al modulo di gestione dei tornelli. Estratto il codice dai messaggi di richiesta, il modulo preleva le informazioni legate al cliente cui è assegnato il tag e, dopo gli opportuni controlli sulla validità delle sottoscrizioni, ne abilita o nega l'accesso inviando il messaggio di eventuale apertura al tornello mittente della richiesta.

Scelte chiave di sviluppo

Esaminiamo brevemente le scelte di sviluppo effettuate per ciascuno dei livelli della struttura Layered enfatizzandone le componenti chiave. Ricordiamo che per l'implementazione del Data Layer si è deciso di utilizzare lo strumento .netTiers, brevemente introdotto nel paragrafo relativo alle considerazioni su Data Layer. Non richiedendo una interazione con l'utente, ma eseguire una continua elaborazione dei messaggi di richiesta di accesso, si è deciso di sviluppare l'applicazione come servizio e quindi di non realizzare alcun Presentation Layer.

Business Layer Il cuore logico è basato sull'utilizzo della libreria fornita con i dispositivi tornelli per permetterne la comunicazione in rete locale. Ogni qualvolta un tornello invia un messaggio UDP di richiesta apertura, il software:

- Estrae l' UID del tag letto dal messaggio UDP
- Preleva le informazioni legate al cliente cui è assegnato il tag oggetto del messaggio
- Esegue tutti i controlli utili a verificare i diritti d'accesso del cliente

- Risponde allo stesso tornello mittente del messaggio abilitando o no l'apertura del braccio meccanico

Piattaforma e dispositivi

Il modulo è stato sviluppato in linguaggio C#, sfruttando la tecnologia Windows Service offerta dalla strumento Microsoft Visual Studio basato sulla piattaforma .Net. Installabile sul DB server non richiede l'acquisto di ulteriori componenti al di fuori dei tornelli. Questi devono avere integrato un lettore RFID ed una scheda di rete Ethernet per la comunicazione su rete locale.

A.7 Analisi dei risultati

Nel capitolo 7 della nostra tesi esponiamo alcuni dati raccolti nei primi due mesi di impiego del sistema. Per motivi di spazio proponiamo in questa sede le considerazioni finali che emergono da tali dati:

- Il sistema di pagamento elettronico sta cominciando a mostrare i primi segni di diffusione. Nonostante il sistema sia perfettamente funzionante, la scarsa familiarità degli impiegati con sistemi di gestione bar/ristorante ha introdotto una certa difficoltà nel garantire il pagamento tramite tag sempre disponibile (questo a causa di un training manchevole da parte del fornitore del software di gestione). Ciò nonostante il numero di transazioni tramite tag ha subito un'inflazione media del 40% ogni settimana, lasciando dunque presagire il meglio per il futuro.
- Le ricariche sui conti virtuali stanno anch'esse aumentando di settimana in settimana e, man mano che i clienti prendono confidenza con il chiosco informatizzato, le ricariche effettuate al botteghino stanno diminuendo in numero, a favore delle ricariche al chiosco.
- Il modulo del sistema più critico in termini di tempo medio di esecuzione dei task, ossia il box office, sta mostrando segni di buona apprendibilità, soprattutto per via delle scorciatoie offerte dalla tastiera durante le procedure più comuni. Un calcolo sui tempi medi di esecuzione della procedura di accesso giornaliero (la procedura più critica in quanto rischia di creare lunghe code all'ingresso) ha evidenziato una discesa dei tempi medi di esecuzione molto promettente, partendo dai circa due minuti registrati nella prima settimana (4-10 maggio) al mezzo minuto della settimana tra il 22 ed il 28 giugno.
- I tag stanno riscuotendo un successo tale da giustificare un ordinativo supplementare a causa degli eccessivi smarrimenti: A livello prestazionale, un buon 20% ha fallito al tornello a causa delle dimensioni ridotte dell'antenna del tag in relazione al campo creato dal lettore del tornello. Una soluzione è attualmente in fase di studio. Su 1200 tag ordinati, due erano illeggibili fin dall'inizio ed uno solo si è rotto fisicamente (silicone spezzato).

- I tornelli hanno processato più di 25000 passaggi senza mai richiedere interventi di assistenza.

A.8 Chiusura del progetto e sviluppi futuri

La chiusura del progetto è stata condotta per mezzo di un meeting con le parti coinvolte ed è ha avuto l'obiettivo di determinare il grado di raggiungimento dei risultati attesi di progetto, operare un'analisi dell'intero progetto dopo il suo completamento e di evidenziare le lezioni imparate dal team.

A.8.1 Raggiungimento dei risultati attesi di progetto

Passiamo ora in rassegna i risultati attesi di progetto evidenziati al capitolo 4 in relazione agli obiettivi di progetto (tabella 4.3), categorizzandoli come RAGGIUNTO o NON RAGGIUNTO.

Miglioramento dell'immagine della MIIB. L'introduzione del sistema ha portato ad un buon numero di commenti positivi tra clienti e media. La notizia ha avuto un certo eco su carta stampata, televisione ed internet. I commenti positivi sono aumentati quando abbiamo chiamato il tag **SeaPass** soppiantando così l'uso del termine braccialetto elettronico che accostava impropriamente il progetto ad un altro (peraltro non realizzato) di dotare i carcerati di braccialetto elettronico. Alla luce del riscontro ottenuto, possiamo considerare il risultato come RAGGIUNTO.

Incremento della qualità dei servizi resi al cliente. Ecco alcune considerazioni sui servizi esistenti:

- **Check in:** la procedura di accettazione è stata resa più efficace ed efficiente dato che il cliente adesso è in grado di scegliere la postazione da una mappa che evidenzia la disponibilità momento per momento. La necessità di un continuo contatto via radio con gli assistenti spiaggia per conoscere la disponibilità delle postazioni è stato eliminato. Il sistema stampa inoltre una ricevuta col numero di postazione assegnato dal sistema cosicché è evitata ogni possibilità di errore umano nella scrittura manuale del numero.
- **Accesso alla struttura:** il cliente operante l'accesso per la prima volta trae spesso vantaggio dall'assistenza di un addetto poichè non vi sono sufficienti indizi visuali a supporto della procedura. Ciò nonostante, una volta compreso il meccanismo, il cliente è libero di entrare ed uscire senza dunque attendere il controllo da parte di una maschera non sempre presente.
- **Acquisti al bar:** avendo fornito un'alternativa ai soldi contanti ed avendo potuto apprezzare un incremento graduale nel numero di acquisti al bar tramite tag, il servizio è stato migliorato.

Date le considerazioni appena esposte, possiamo considerare questo risultato come RAGGIUNTO.

Controllo degli accessi automatico ed efficace che prevenga ogni abuso. L'introduzione dei tornelli ha avuto un impatto positivo considerevole sul controllo degli accessi. Nei giorni di maggiore affluenza era pressoché impossibile per la maschera controllare tutti i clienti (autorizzati peraltro ad un continuo viavai da una parte all'altra dei punti di accesso). Il funzionamento del tornello fornisce invece un controllo puntuale, rigoroso e continuo, prevenendo così qualsiasi abuso. Il risultato atteso è stato pertanto decisamente RAGGIUNTO.

Diffusione di un sistema di pagamento proprietario all'interno della struttura. Lo strumento di pagamento è disponibile al bar interno alla struttura. Come i dati presentati al capitolo 7 mostrano, dopo un'iniziale partenza rallentata dovuta a diffidenza da parte dei clienti, un incremento nell'uso di questo servizio è riconoscibile. Consideriamo quindi questo risultato come RAGGIUNTO.

Raggiungimento di una solida base di informazioni su cui basare scelte di breve, medio e lungo termine. Nello sviluppare il nostro progetto, ci siamo curati di fare in modo che il sistema raccolga qualsiasi tipo di informazione possa rivelarsi utile per le decisioni strategiche della compagnia. Oltre alle informazioni su clienti e sottoscrizioni, il sistema raccoglie dati circa accessi, servizi, acquisti al bar, malfunzionamenti e molto altro ancora. Visto che i primi due mesi di raccolta dati hanno già prodotto informazioni prima inaccessibili per la MIIB, possiamo considerare il risultato RAGGIUNTO. interesting analysis from only two months of collecting data, we can consider this result as **ACHIEVED**.

Controllo efficace sulla liceità della fruizione dei servizi offerti dalla struttura. Questo risultato è legato all'implementazione di un meccanismo di ronde periodiche supportate da palmare. Visto che il modulo palmari non è stato progettato ma non implementato per vincoli di tempo, questo risultato deve essere categorizzato come NON RAGGIUNTO.

A.8.2 Lezioni imparate

Il meeting svolto ha portato alla luce le seguenti lezioni imparate dal team:

- un'analisi di mercato preliminare sui prodotti disponibili può aiutare a risparmiare tempo nella selezione dei fornitori;
- sovrastimare almeno del 20% i costi di progetto delineatisi durante lo studio di fattibilità;
- è molto difficile conciliare le attività di Project Management con le fasi di analisi, progettazione e sviluppo;

- non sprecare tempo cercando di fare le cose perfette se hai risorse bastanti per farle bene;
- considera sempre del tempo supplementare per i fornitori extra-europei, e non per i tempi di spedizione - in alcuni casi anche inferiori a quelli mostrati da fornitori italiani - ma più che altro per il tempo necessario a concordare le caratteristiche della fornitura ed a far partire la produzione/assemblaggio;
- meglio dedicare del tempo supplementare alla raccolta dei requisiti visto che ogni cambiamento dei requisiti ha un impatto sulle tempistiche crescente man mano che il progetto procede;
- è importante avere incontri di persona per la raccolta dei requisiti anziché comunicazioni a distanza;
- deliverable documentali troppo grandi saranno difficili da valutare per il cliente e la qualità del riscontro ne farà le spese;
- affrontare lo sviluppo attraverso una decomposizione per layer delle responsabilità consente di parallelizzare il lavoro ed ottenere un'integrazione efficace;
- assicurati di stare prevedendo il corretto numero di risorse in relazione con le dimensioni del progetto. Un numero insufficiente può avere una grande varietà di conseguenze negative sul progetto.

Bibliography

- [1] Charles Walton - The father of RFID - <http://radio.weblogs.com/0105910/2004/06/14.html>
- [2] A Small Piece of RFID History - <http://www.rfidjournal.com/article/articleview/3315/1/366/>
- [3] Wikipedia - http://en.wikipedia.org/wiki/Use_case
- [4] http://en.wikipedia.org/wiki/Entity-relationship_model
- [5] Application Architecture Guide 2.0 - Designing Applications on the .NET Platform