Universidad de Oviedo

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

GRADO EN INGENIERÍA EN TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

ÁREA DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

TRABAJO FIN DE GRADO

DEVELOPMENT OF RFID APPLICATIONS FOR MANAGEMENT AND TRACKING ASSETS AND DRUGS IN HOSPITALS

AUTOR: JACQUELINE FRANSSEN
TUTOR: YURI ÁLVAREZ LÓPEZ
COTUTOR: GUILLERMO ÁLVAREZ NARCIANDI

FECHA: 16 de julio de 2018

Contents

List of Figures

Abbreviations

RFID Radio Frequency Identification

RF Radio Frequency

NFC Near Field Communication

IT Information Technology

HIS Hospital Information System

RIS Radiology Information System

LIS Laboratory Information System

EPC Electronic Product Code

WORM Write Once Read Many

MVC Model View Controller

MVVM Model View View Model

WARD Wisely Aware RFID Dosage

MIMS Mobile Intelligent Medical System

LF Low Frequency

HF High Frequency

SSL Secure Sockets Layer

TLS Transport Layer Security

FDA Federal Drug Administration

PZN Pharmazentralnummer

SMLE Single Logical Message Exchange

SARS Severe Acute Respiratory Syndrome

LBMS Location-based Medical Service

TMUH Taipei Medical University Hospital

ERP Enterprise Resource Planning

ITC Information and Communication Technologies

HL7 Health Level 7

DICOM Digital Imaging and Communications in Medicine

LAN Local Area Network

SOA Service-oriented architecture

WLAN wireless local area network

IrDA Infrared Data Association

PC Personal Computer

SOAP Simple Object Access Protocol

API Application's Programming Interface

WfMS Workflow-Management System

ASP application-service-provider

FIFO first in first out

ESB Enterprise Service Bus

SWOT Strengths Weaknesses Opportunities Threats

CEP Complex Event Processing

SME Small and Medium-sized enterprises

SQL Structured Query Language

JSON	JavaScript Object Notation
XML	Extensible Markup Language
DBMS	Database Management System
OSN	Online Social Networks
BSON	Binary Structured Object Notation
NAS	Network Attached Storage
SAN	Storage Area Network
CLI	Command Line Interface
VM	Virtual Machine
WPA2	Wi-Fi Protected Access 2
CPU	Central Processing Unit
RAM	Random Access Memory
PCI	Peripheral Component Interconnect
GB	GigaByte
RDBMS	Relational Database Management System
HUCA	Hospital Universitario Central de Asturias
TC	Technetium
SPECT	Single-photon emission computed tomography
SESPA	Servicio de Salud del Principado de Asturias
LOS	line of sight

Introduction

The following Bachelor Thesis describes the development of a mobile hybrid application which can be run both on Android and iOS devices as well as the physical layer hardware based on RFID technology. The developed system (both hardware and software) is devoted to be used for tracking and management of medical assets and drugs in hospitals. Compared to other existing systems, the given Bachelor Thesis includes the development of 'real-time' synchronization between client and server, implemented by Socket.IO. Thus, a synchronous solution approach is pursued in order to provide real-time actions concerning drugs and medicines administering.

The project aims to develop a RFID-based system for management and tracking applications. In particular, the proposed system is devoted to be deployed at hospitals, to ensure a better control of the medicines and medical tools that are periodically administered to the patients, reducing the probability of missed or wrong medicine administration. The proposed system is formed by a set of RFID antennas that will conform the coverage area where items tagged with RFID tags will be read. These antennas and the RFID reader will be connected to a database that will contain the information of the items to be monitored. The project covers topics of great interest for medical and Information and Communication Technologies (ITC) companies, such as IoT and e-Health.

1. Use and scope of RFID Technology

The following chapter discusses the reasons supporting the choice of Radio Frequency Identification (RFID) technology to address the technical challenge presented in this Bachelor Thesis. First, the most relevant state of the art applications are described in section 1.1. After that, the section 'Aim and Scope' 1.2 introduces the limits of the RFID technology which should be considered during the deployment. Finally, several RFID applications and companies which provide RFID based solutions are analyzed using the Strengths Weaknesses Opportunities Threats (SWOT) method to show the benefits and threats.

1.1.- Motivation

Concerning the organization and management of medical devices or patients in a hospital, Ajami and Rajabzadeh propose several technical solutions to address some of the most frequent logistic and operative problems.

1.1.1.- Decision Making

First of all, concerning decision making, e.g. about the correct treatment of a severe illness, many physicians are stumped for an answer or their opinions are divided. To enable a rapid diagnosis and to improve the patient's health status, 'smart healthcare' [1] would help a lot. 'Smart healthcare' includes RFID tags which are equipped with sensors to ensure the effectiveness of a medical treatment. This accelerates the treatment process a lot. Furthermore, patients or hospital beds equipped with RFID tags make it easier to identify and manage the amount of patients as well as the workflow.

1.1.2.- Internal Communication

Secondly, poor communication between nurses and physicians deteriorates medical supply. For instance, if a nurse notices that a patient needs more tranquilizer because he became very nervous, she has to tell the doctor to dose the patient with the correct amount. But often, a physician is occupied with another patient. Therefore, communication problems between the staff arise. Thus, inadequate patient monitoring emerges. It should be added that sometimes there is the risk of misidentification of patients. To explain the last point, one can think of this easy example: At the urology department are two elder patients, Paul Schmitt and Jochen Schmitt. They are not brothers or related to each other and suffer from different types of illness. Paul suffers from kidney insufficiency whereas Jochen suffers from prostatic lithiasis. The first one needs a dialysis every day whereas the second one needs a radiosurgery. Because both patients are unable to walk themselves, nurses and clinical staff have to bring them to the particular treatment room. The problem should be easy to understand, both patients have the same surname but need completely different treatments. If the treatments would be commuted, their health status would deteriorate and they might die because of the misidentification. That is why they should be identified uniquely by for example an RFID system (wearing RFID bracelets).

1.1.3.- Production process

To give another example of the successful deployment of RFID solutions, Tamm and Tribowski [2, p.110 ff.] outline the company 'Gerry Weber'. Next, some benefits of the RFID application will be explained. First, count and identification processes of goods could be accelerated by using the RFID technology. Second, both the electronic article surveillance and RFID minimize costs and time. Third, the delivery quality was improved and mistakes were reduced and sometimes avoided. Fourth, the logistic was improved by the transparency of stock. Lastly, the existing heterogeneous system can be controlled more easily.

1.1.4.- Investment possibilities

Another important point for hospitals is the budget and their possibilities to investigate in new technologies which makes the enrollment of a new RFID system more challenging. Furthermore, clinical staff and phycisians have to be introduced into the new technologies. Not only the human factor plays a significant role but also the existing systems, such as the Hospital Information System (HIS), Radiology Information System (RIS) or Laboratory Information System (LIS). If a new identifying system or software shall be integrated into a hospital or healthcare institution, it has to be deployed suitably to the existing system architecture. To achieve the last point, Ajami and Rajabzadeh [3] recommend starting with small RFID projects and mention countermeasures to increase the acceptance of such applications by healthcare institutions. To give an example, the regulations to protect patient's privacy should be mature to achieve more institutional support.

1.1.5.- Medication Administration System

Next, a few applications which have a positive impact will be described briefly (see also [3]). Firstly, Ajami and Rajabzadeh describe a Medication Administration System which automatically verifies medication and generates the corresponding prescription. There are multiple intents of developing an Administration System, such as preventing human errors (like for example mislabeling of tissue specimens in gastrointestinal and colorectal surgery endoscopy units). To avoid these errors, an initiative of developing an RFID application for specimen bottles was started. The aim of this initiative was to create a paperless pathology requisition system which correctly confirms boths the endoscopy nursing staff as well as the endoscopist for each specimen bottle. After deploying the application, specimen-labeling errors were significantly reduced.

1.1.6.- Wisely Aware RFID Dosage

Another RFID system, called Wisely Aware RFID Dosage (WARD) system should prevent the risk of medication errors triggered by medical staff. It is based on an

integrated barcode and RFID tags which should demonstrate effective and safe patient care environment. Not only the correct dosage can be controlled by RFID but also medical staff. The following paragraph will describe the Mobile Intelligent Medical System (MIMS) which includes a mobile nursing care system using RFID technology. There are many implemented functionalities in the MIMS, such as the tracking of patient's vital signs across various locations and in different medical facilities. The vital sign monitoring enables medical staff to watch critical ill patients carefully and permanently and reduces the risk of serious harm resulting from slow provision [3]. Moreover, it offers alarming services in case of emergencies and can always be taken everywhere. Behind the frontend, a rule-based clinical decision supports medical staff and the mobile nursing environment. Last but not least, MIMS has been extended to most medical domains and has been integrated into other HIS.

1.1.7.- RFID applications in hospitals: A case study

In [4], Wang et al. describe a case study of implementing a RFID system in a Taiwan hospital in the year 2003. The project was named Location-based Medical Service (LBMS) and performed at the Taipei Medical University Hospital (TMUH). In the next section, the development strategy, device management as well as the value generation which were important for developing the LBMS will be explained. Referring to a widely spread disease, called Severe Acute Respiratory Syndrome (SARS) in 2003, the authors Wang et al. discuss the effectiveness of applying RFID in hospitals to prevent further infections (e.g. of patients or medical staff). They mention several challenges of implementing RFID systems in hospitals, for instance user or physician resistance, investment problems as well as technical, clinical, organizational and professional resistance. Nevertheless, some hospitals initiated (with subsidies from the Taiwanese government) preliminary RFID projects as early as October 2003 and achieved significant results.

To give a basic introduction into the existing IT infrastructure at the TMUH, the following paragraph will mention the existing systems of the hospital. TMUH has an integrated HIS that complies with several healthcare standards, such as Health

Level 7 (HL7), Digital Imaging and Communications in Medicine (DICOM) [4, p.3 ff.]. Furthermore, the system consists of a LIS, RIS and according to Wang et al. most of the patient's medical records are digitalized. With regards to the development and the reasons for using LBMS, the authors claim to build a system that could detect and track potential SARS cases. Besides, medical knowledge and practice should form the basis and core for developing the system. The RFID technology was considered as a tool to support medical practice. In the end, the system should reflect medical assumptions.

Wang et al. describe a basic workflow with four steps of the LBMS: Initially, all data should be stored in a positioning database which is connected to the existing vital information databases (of the HIS). In the second step, the system automatically retrieves patient medical records from the HIS and runs an inference engine (called 'Rulebase'). 'Rulebase' judges whether there was an infectious event or not. If there was a infectious event, the system detects this in a third step. As a consequence (step four) of the detected event, a message is sent immediately to the relevant personnel through an alarm (email and sms).

The LBMS can be extended and used in other contexts, like e.g. for precious equipment tracing, in-patient medicine auditing, new-born baby and mother identification or to legitimate drug control. Wang et al. were supported by the Taiwanese government which approved their plan and granted money. Since the LBMS should be released as a hospital-wide system, the development required expertise and knowledge from different domains, including medicine, RFID technology, IT systems development, telecommunications and systems integration. Actually, three parties were involved: TMUH, Lion Information Inc. and an advisory group [4, p.4] which consisted of professors who emerged the technology and made academic contributions (algorithms). Since the hospital decided that the system should have active real-time position-tracking, temperature taking and monitoring abilities for tagged patients, the developing team chose 916.5 MHz UHF active tags (see Chapter 2, RFID tags 2.1.1.1) to reduce the risk of staff infections.

Reaching an adequate system integration without loss of performance, functionality and security was a big challenge. With the use of a field generator, a small tag wake-up device that communicates directly with the reader, the real-time communication should be realized [4, p.4]. The generator periodically turns on and calls tags for a specific time. There are three different types of generators: Normal, floor and area generators.

Furthermore, Wang et al. bring up the challenge of the entire device management [4, p.5] with the purpose of collecting and transmitting reads that are as complete and clean as possible. Realizing a complete device management was limited by compartments, rooms, walls and doors because of their building layouts and materials which interfere with radiowaves. Besides, the balance between accuracy requirements and investment costs have to be maintained. Moreover, unauthorized removal of tags has to be managed carefully, since there might be some patients who try to take off their RFID wristband. In this case, an additional alarm has to be designed. Basically, the design and deployment of RFID devices depend on the environment and the context in which they are used.

Not only the device management was challenging but also the data management as Wang et al. mention. The authors describe two general problems of data management in their RFID system. On the one hand, there will occur intermittent and unreliable reads. These can be compensated by developing algorithms to process missing and incomplete reads. On the other hand, there will be generated high-volume data in a very short time. To prohibit this, the data should be filtered by algorithms and only the necessary data should be transmitted. For instance, if a tagged patient exceeded the present degree of 0.5°C, the data associated to this patient would be transmitted. To come to a conclusion, data management is tied to medical knowledge and practices which can substantially reduce the volume of data to be handled. As a result, meaningful information for decision making will be generated.

Besides the LBMS project [4, p.2 ff.], Wang et al. depict some existing RFID applications. To give an overview of the usual hospital applications until 2006, Wang et al. describe applications for tracking and managing equipment such as wheelchairs

or portable heart monitors. Moreover, trials on tagging patients, staff and equipment in rooms were conducted in several hospitals. Besides, the Washington Hospital Center (Washington D.C.) deployed a RFID system to track the status and the exact location of patients, staff as well as the essential equipment.

During realization, building an RFID infrastructure together with the middleware and the impedance-matching of the RFID system and the current systems (e.g. Enterprise Resource Planning (ERP) systems) was important. Actually, in order to get along with the mentioned solutions, a strong team work (involving people from IT and business departments) and project management should be included.

Since RFID allows wireless storage and automatic retrieval of data, there is an 'ecosystem' of companies trying to develop a platform to support RFID development and applications. Besides, the variety of existing systems in hospitals, Wang et al. mention three mayor technical challenges accomplishing a RFID system. First, the non-line-of-sight reading might be a challenge since there are various types of tags and the frequencies influence the range of signal. Second, handling the serial numbers can be difficult but it could be coped with setting a primary key to each tag which synchronizes with an existing database (see Chapter 3, 'Used platforms and technologies' 3.2). The third challenge is to deal with the real-time data and to synchronize these seasonably. To deal with that, the use of NoSQL databases makes sense and will be discussed in Chapter 3 2.2.

Finally, Wang et al. evaluate RFID as an infrastructure technology which allows companies to capture data about objects and individuals moving in the real world [4, p.7]. In addition to that, the authors recommend that organizations should think carefully how to change business processes to reap the benefits of RFID. By naming benefits of RFID, Wang et al. refer to the improved efficiency, patient safety and reduced medical errors which can be very extensive and expensive nowadays [4].

1.2.- Aim and Scope

Ajami and Rajabzadeh [3] mention three important purposes of RFID technology. The first purpose of using RFID is to improve the tracking of objects. It is mainly used to follow products through a specific supply chain or to follow medical devices an drugs in the clinical workflow. There is also the possibility to track a product to a particular patient or to identify clinicians who administered medication to patients.

The second purpose for which RFID technology is appropriate is the inventory management (see section **??**). Inventory Management is significant for managing items of an organization, like in a hospital. There are many complex processes where information about the location, time and the amount of material is necessary (e.g. towels, duvet covers). The third and last purpose of RFID technology, mentioned by Ajami and Rajabzadeh [3], is validation. Using RFID to identify and validate data is an effective method for ensuring the quality of a hospital or healthcare setting. It ensures that the patient being treated is the right patient.

1.2.1.- RFID and Mobile Computing

Concerning the development of mobile applications in context of the RFID technology, the following paragraph discusses fundamental definitions of mobile computing as well as general architecture patterns, like e.g. Service-oriented architecture (SOA). To start with, Hanhart declares in his book [5, p.9 ff.] the term 'mobile computing' as following: '[...] all processes, activities, applications in a company which are proceed by mobile technologies. The company's staff gets access to data and applications (independently from location and time). The focus is set on man-machine-communication [...]' [5, p.9 ff.].

With respect to mobile applications which are connected to RFID systems, Hanhart mentions the term of 'Smart things' or 'Embedded systems' which include physical objects, extended by the RFID sensor technology and which are networked to each other. To achieve some understanding among all readers, Hanhart depicts three types of wireless communications technology [5, p.12-13]. Firstly, there are mobile

communications which consist of a service provider and several mobile devices. The service provider transmits the speech and data from and to mobile devices through a wireless network.

Secondly, Hanhart notices wireless local area network (WLAN) which enables accessing to a company's network. To give an example, there are many WLAN hotspots in hotels and airports, which can be accessed simply. Thirdly, there are 'wireless personal networks' which connect terminal devices with peripheral devices within small ranges. For instance, Bluetooth or Infrared Data Association (IrDA) are used to transfer data over small distances. Not only Bluetooth and infrared light are used to transfer data, but also ZigBees and Near Field Communication (NFC) is used in many cases [5, p.12-13]. Hanhart describes NFC as a newer technology which consists of an active and passive unit. Actually, NFC is only used for connections of a few centimeters, e.g. in consumer electronics. The active unit can be kept in the mobile because it is very small and the data rate amounts to at least 424 kbit/s.

1.2.1.1.- Basis functions of mobile computing and RFID

Concerning the use of mobile applications and RFID, one can distinguish between five general scenarios [5, p.13 ff.]. First of all, since RFID enables wireless connection and detection of several items in our environment, it can be used to access mobile applications through the RFID signal. For example, staff can use mobile devices to access (via RFID) business applications and to implement transactions. Secondly, as already mentioned in further sections above, RFID has the main purpose of identifying objects. In combination with mobile devices, there can be established applications which can both identify users and objects.

As a third scenario, mobile RFID applications the capture of statal and environmental data. By using sensors, the continuous capture of data is possible. Moreover, smart objects and mobile devices can transform the information directly into actions or convey the data to a central system (or database). As a fourth scenario for using RFID in the context of mobile computing, Hanhart [5, p.13 ff.] remarks the possibility to locate precisely objects and users. Besides, the detected positions

can be used to display contextual information or to direct to procedures in backend systems. Last but not least, the purpose of sending notifications in time, can improve and prevent several emergency situations (e.g. in a hospital). Since objects can send notifications e.g. when reaching a certain state, users get information everytime and everywhere.

1.2.1.2.- Constraints of mobile applications and RFID

Developing mobile applications with the RFID technology not only brings advantages, but also challenges when facing for example the user's needs and requirements. Hanhart indicates some 'constraints' of mobile applications and RFID [5, p.16 ff.] which will be depicted soon. Firstly, the user interface should be considered. In case of healthcare applications which can be used by nurses, staff and physicians there should exist various user roles and rights. Thus, each user needs his specific interface and only a few (or one single) users are able to see all information of a patient.

As a further matter, the mobile application should meet the requirements of connecting quality and service quality. This indicates the exact adaption of a service, specified in the requirements specification document. By the same, the challenge of computing capacity in the developer team should be attended. Furthermore, both principal and agent should be aware of the needed development time to realize all required features (which depends on the size and skill level of the developers). In addition, the technical resources, like e.g. memory capacity and energy supply represent another challenging factor. To deal with the last mentioned challenges, it should be favourable to have some sponsors which can support the project.

1.2.1.3.- Service-oriented architectures

To cope with the constraints and challenges during the development of a mobile RFID application, Hanhart describes SOA [5, p.31 ff.]. SOA is a multilayered, distributed information system architecture which encapsulates parts of an application into business-like services, considering design principles to enable a simplified process integration [5, p.32]. A service can be seen as an abstract software element or interface

which provides standardized access to application functions of other applications through a network [5, p.32]. There are four general design principles with respect to the development of SOA applications.

The first design principle is called 'Orientation of Interfaces' which means that the service interfaces abstract implementation from the user's view. What is more, each service has a stable interface which is technically and functionally defined by its metadata. The second design principle of SOA is 'Interoperability' which can be assured by implementing technical and functional standards. This enables the interoperability of a services and its usage in different contexts. The third design principle is called 'Autonomy and Modularity' which signifies that SOA restructures the applications architecture into autonomous subsystems (domains and services). The aim is to increase cohesion in one system and to minimize the linkage between its subsystems. The fourth design principle of SOA is 'Orientation of needs'. This implies that all services should be oriented towards business objects and process activities in order to provide an approximately granular, functional definable output.

1.2.1.3.1 Domain Architecture The Domain Architecture is a conceptional foundation of SOA. It reveals duplicates in an existing application architecture. Moreover, it is a fundamental decision foundation for service characteristics. Last but not least, it gives a list of possible service candidates which support systematic integration, development and usage of services.

1.2.1.4.- Service-oriented proposal of architecture

In the previous section, the general characteristics of SOA have been discussed. But in which context SOA applications are regularily used? And how is SOA realized? To give answers to these questions, next the purpose and integration procedure of SOA applications will be explained. First of all, the conditions and prerequisites for an economic realization of diverse solutions based on mobile computing and RFID are simple and flexible integration into existing system architectures [5, p.133 ff.]. The classical integration of mobile terminal devices involves different middleware

components and functions which can have different varieties of architecture (with advantages and disadvantages).

Modern application systems are multilayered, for example 3-tier/n-tier-architectures. The three tiers include the client (implemented software-components), middleware (necessary components to connect client and backend) and backend (business data and functions) tier. The term 'mobile middleware' refers to the extension of the classical client-server architecture with the aim of improved scalability and administration.

Not only a mobile middleware is needed for an appropriate integration into SOA, but also a RFID middleware is needed. The following paragraph will mention some functional requirements to this specific middleware. First of all, transformation functions are needed to convert RFID raw data into useful business process data. This includes filtering of errors as well as harmonizing the data formats of different device manufacturers. Secondly, there are configuration functions needed for monitoring and controlling the RFID infrastructure. As follows, a configuration function is able to identify readers, manage configuration data, monitor device functionality and ensure security.

1.2.1.4.1 Classic Integration Mechanism As a third type of integration, concerning mobile computing and RFID, the integration of embedded devices should be considered. Basically, there are three different tiers: Field tier, Automation tier and Management tier. The first tier, Field tier includes the physical connection of sensors, actuators and control units. The second tier, Automation tier, consists of control units which undertake automatic monitoring and processing functions, like for instance the control of temperature. The communication units connect the control units, programming units and the management tier.

The third tier, called Management tier, subsists of monitor and control systems and visualizes them to the operator. Additionally, the Management tier delivers software responsive over proprietary interfaces and can be seen as data handling unit. To conclude, the Automation tier can be seen as the communication unit and middleware

between Field tier and Management tier. Further, Automation and Management tier can connect to third-party-applications via data interface units.

1.2.1.4.2 Modern Integration Mechanism The last paragraph dealed with integration mechanisms on the real old way. Recently, there are newer technologies and possibilities to realize an integration, such as web services and the emerging communication protocol Simple Object Access Protocol (SOAP). In context of web technologies and services, there comes up the term Enterprise Service Bus (ESB) [5, p.141 ff.] which provides a standardized interface and communication layer. An ESB is a consistent integration architecture which defines standards as well as central services and provides them for software development, publication and usage.

Regarding the integration of mobile applications, the ESB [5, p.143], there are differentiated two types of integration scenarios: a) the mobile client calls directly the provided services from application domains or b) the client keeps communicating with the services via its mobile middleware. If the client wants to call a service directly, he has to implement its Application's Programming Interface (API). Besides, for service call or invocation, the standardized interface technology has to be used.

In the matter of integration of RFID systems and embedded devices, the middleware of RFID systems uses the provided services of the application domains, provided by the ESB. In addition to that, mobile applications can be integrated both online and offline [5, p.146 ff.]. Hanhart is using the term 'online' in context of saying that the mobile device runs the user interface.

This indicates that the mobile middleware prepares contents of application (for prompt). After that, the application accesses the backend or invoked app through a service which uses the application's functionality via services. In contrast to that, Hanhart uses the expression 'offline' in order to say that the application is running on the client. Here, the mobile application acts as an invoked application and synchronizes its data through the mobile middleware with the backend. The central management of process states (on the client) and synchronization with services are realized through the middleware directly.

1.2.1.4.3 Capabilities of SOA, mobile applications and RFID To explain the capabilities of the above discussed technologies and architecture, Hanhart brings up 'Reuse', 'Isolation of Domains' and 'Easy implementation'. Reuse is generated by consistent functionalities of interfaces. Isolation of domains is produced by separation of concerns and core data concepts. Easy implementation refers to cross-domain workflows or taskflows. Concerning possible usage scenarios, Hanhart mentions some examples [5, p.207 ff.], like event-driven process management which includes the automatic capture of events. Or, if it comes to the control of several processes and activities in a company, these activities can be executed by using mobile phones or tablets. Additionally, they will record the encountered states.

Concerning the 'ecosystem' and exchanges in it, Hanhart talks about some upcoming challenges [5, p.212 ff.], like the realization of mobile solutions. It is important to consider the high costs of hardware and software. In addition to that, the complexity of integration of solutions and backend systems has to be assumed very well. Lastly, there should be staff who configures and operates the mobile devices.

1.2.1.4.4 Examples of SOA applications Hanhart mentions several use cases of mobile applications and RFID. In order to depict two of his examples, the following paragraph will explain firstly the case study: 'Fraport AG' [5, p.39 ff.] and secondly a Workflow-Management System (WfMS) [5, p.204 ff.]. To start with, Fraport AG is a company which is simulataneously owner and operator of Frankfurt airport. By including the RFID technology and mobile devices, several solutions have been realized in processes for mobile support of staff and in use of real-time data. The scope contained the maintainance of fire dampers and mobile support of loadmasters during loading respectively unloading and mobile capture of booking of goods input and goods issue in stock.

The second example Hanhart notices, is a WfMS [5, p.204 ff.] which is able to control processes, e.g. in companies. To realize a WfMS, the workflow client needs to be installed on the mobile device. Following, RFID systems and embedded devices are

able to report events to WfMS and can trigger or control processes on the workflow integration layer.

To give an outlook of the possible extensions of mobile applications and RFID, Hanhart explains some fundamental concepts [5, p.208 ff.]. The first concept is called 'Emotion-Silent Process' and is based on sensors, actuators, artificial intelligence which includes the learning from given data and derive several activities. In the given usage scenario (which refers to a hospital or retirement home), sensors firstly detect the movement of a elder person.

After the detection step, the sensor's information are connected to further data (e.g. time) with the help of pattern detection algorithms. Finally, the aim of the 'Emotion-Silent Process' is to detect and prevent emergency situations by sending an alarm to responsible persons (like physicians or nurses). Furthermore, the process is called 'Emotion-Silent Process' because of the 'silent' protection of people which should obtain the independance of these people.

A second usage scenario which is noticed, is called 'Velocity-Outtasking and Application Outsourcing'. Based on the spread of web services in cross-company cooperations either external access to single tasks and functions ('outsourcing') or to whole applications ('application outsourcing') can be accelerated. The term 'Outtalking' is used to talk about web services of external providers which can be incorporated to the own service repository and are centrally available for further usage. In contrast to that, 'Application Outsourcing' refers to externally obtained applications (from application-service-provider (ASP)) which can be integrated through the company-internal ESB into the system landscape.

1.2.2.- Analysis of RFID applications and their use

Often, there not only exists one ideal solution for implementing RFID applications, but multiple applications. In order to decide whether the proposed solution is the appropriate one, the SWOT method has been established [2, p.47 ff.]. SWOT is the acronym for strengths-weaknesses-opportunities-threats and can be seen as the epitome

of a popular instrument for self-analysis of decision makers. The analysis itself can be divided into two comprising steps: In order to achieve the objective with the given system, its strengths and weaknesses (with an internal origin) are identified in the first step. After that, the system's environment is analyzed by ascertaining external opportunities and threats.

To give an example of the SWOT analysis, Tamm and Tribowski [2, p.47 ff.] consider two perspectives on RFID applications: The 'Enterprise Perspective' as well as the 'Political Perspective' which will be described soon.

From the enterprise's point of view, the strengths of RFID systems are the optimization of operational processes and the reduction of costs. Further, many enterprises improve their transparency because of the raised quality of data (improved decision making). In contrast to that, there are weaknesses of RFID applications from the view of enterprises, e.g. the challenge of integrating them into the IT, the physical as well as the organizational integration of processes. With respect to the environmental analysis of enterprises, there occur some opportunities, like for example new business models which are based on RFID. Or, for instance there will arise innovative business partners which cause a general willingness to cooperate with them.

On the opposite, there also might occur some threats in the context of RFID applications in enterprises. To give an example, there might appear an asymmetrical cost-benefit in the value chain. Moreover, any application standards are missing to establish a cross-company infrastructure [2, p.47 ff.].

From the political perspective, the strengths of RFID applications are receptive RFID users, effective manufacturers of the technology, exploratory infrastructure and strategic projects. In contrast to that, the weaknesses of the mentioned systems could be the risk of investment at Small and Medium-sized enterprises (SME), an inadequate alignment of the technology in Europe and missing application standards. As external opportunities can be seen the high potential to gains in efficiency, the emerging new employments, the market share for european technology manufacturers and the resulting data privacy technologies. On the other hand, Tamm and Tribowski

[2, p.47 ff.] mention political threats which might exist in the german point of view. For instance, there might occur competitions in technology development with the USA and Asia, rising discounters beyond Europe, missing global interoperability as well as a missing consensus of sociopolitical problems.

According to Tamm and Tribowski [2, p.95 ff.], the highest potential of the RFID applications is the cross-company deployment of RFID because of the raised visibility in a cross-company solution. Generally, the benefit of network technologies depends on the deployment of the application in the network as well as on the number of partners which are integrated into the application. Nevertheless, cross-company RFID solutions enable cost reduce due to the distribution of costs to multiple stakeholders and because of the multipurpose of their transponder.

1.2.2.1.- Process Model

When establishing a RFID system in a company or building a cross-company infrastructure, in practice, process models are used to roll out. Process models divide a process into definite phases [2, p.59 ff.] which can be named as the following: 1) prephase, 2) analysis phase, 3) draft phase, 4) implementation phase and 5) adoption phase. In the succeeding paragraph, each phase will be depicted briefly.

Firstly, the prephase includes the definition of determine aims, the project team, requirements to the application, funding models and approach. After that, in the second phase (analysis phase) information about all stakeholders is collected. Afterwards, all used IT systems are analyzed in order to integrate the RFID technology correctly into existing applications. Besides, the technical infrastructure is analyzed to find out the technical characteristics of RFID. In addition to that, the functional requirements are defined.

In the third phase, the draft phase, the target process is the documentation of the functional specification document which is based on the specification book. Next, during the implementation phase, the software is developed and tested, hardware installation, configuration, tests are performed. Every action as well as every result is

documented and later used as training material. All in all, this phase aims to implement an operational solution which can be adopted to the existing systems.

Before the last phase, qualification measures for all stakeholders are performed in order to check the appropriateness of the application to the specified requirements of phase three. In the last phase, the adoption phase, the developed application is adopted and released. With the adoption, the maintainance phase of the application starts.

To conclude this recommended process model, one should consider that it is only a 'model' described by Tamm and Tribowski [2, p.59 ff.]and project-specific variables like e.g. number of team members or ressources can influence each of the stated phases.

2. Functionality of RFID technology, NoSQL technology

The second chapter is divided into two parts: the functionality of RFID technology and NoSQL technology. In the first part, several characteristics of the used RFID technology and the components of a RFID system (see 2.1) are depicted. Concerning the security and privacy of RFID systems, section 2.1.3 refers to some useful methods against threats. After that, in the second bigger part 2.2, some general information about NoSQL is given and compared to SQL technologies. Furthermore, the term 'Big Data' is introduced and discussed briefly in section 2.2.2. Finally, a sample project which uses various NoSQL technologies will be introduced in section 2.2.3.

2.1.- RFID technology

According to Ajami and Rajabzadeh [3], RFID technology is capable of an automatic unambiguous identifiation without being placed in the line of sight (LOS) of their objects. The data between RFID tags and readers is transmitted through radio waves. An early RFID-like technology was used as early as in 1940's to identify airplanes during Second World War. Today, it is used in several different areas, like for example in manufacturing, supply chains, agriculture, transportation systems, healthcare services etc.

2.1.1.- Components of an RFID application

Ajami and Rajabzadeh [3] mention five main components existing in a RFID system. Firstly, there is the RFID tag attached to an object ensuring its unique identification. Secondly, the RFID reader detects each tag and generates a response: an electromagnetic wave is sent back to the RFID reader from the detected RFID tag, conveying the RFID information onto that wave. In order to detect tags, one or

more antennas have to be connected to a reader. Thirdly, in every RFID system has to exist a communication infrastructure which enables the interaction of readers and tags through an Information Technology (IT) infrastructure. Lastly, to enable users to connect to the RFID infrastructure and to control its modules, there has to be established an application software including an user interface and a backend service (e.g. database).

2.1.1.1.- RFID tags

When it comes to the variety of RFID tags, three fundamental types are distinguished: Active, semi-active and passive RFID tags [1] which all consist of an antenna, a microchip and packaging. Active RFID tags consist of a microchip and have their own power source. As a characteristic, they are more expensive than the other two types. After that, semi-active tags or also called 'hybride' tags have their own power supply which is only used to support the microchip. The transmission or communication between semi-active tag and reader is implemented by using the power of the reader's field. Finally, the passive RFID tags do not consist of a power source and only work in the reading range of the reader. They harvest their needed energy from the electromagnetic field of the reader and are cheaper than active tags. Moreover, passive tags are lighter than active tags and provide a long-lasting service. In constrast to active tags, passive tags are limited in their read range and functionality.

The 'EPCglobal', an industrial consortium, distinguishes between five classes of RFID transponders (tags) [2, p.15 ff.]. The first three classes include passive tags which have no own energy or power supply whereas the last two classes are used to identify active tags. Particularly, the 'Class 0' signifies that the serial number is written during production process. The 'Class 1' means that a transponder can only be labelled once. 'Class 2' means the tag is rewritable, e.g. the serial number or further data can be rewritten.

'Class 3' represents the tags which have their own internal battery for a microchip but whose data exchange (sending and receiving information and messages) is supported by the reader's energy. When it comes to the last two classes, Tamm

and Tribowski remark the purpose of reassessment, aggregation and transformation of RFID data. Actually, these active tags are no 'real' RFID transponders but 'telemetry transmitter' because they do not influence the electromagnetic field of the reader and do have their own electromagnetic field. In particular, 'Class 4' refers to tags which have their own power supply which is used for the microship and data exchange. Furthermore, they cannot communicate with passive transponders. The final 'Class 5' appoints to tags which can also communicate with passive transponders.

According to Henrici [1], the memory capacity of passive RFID tags can vary from single bits to kilobytes which is not much. As a recommendation, an external database to store tag-specific data should be used. For instance, a memory of 12 bytes is very common to store Electronic Product Code (EPC). Concerning the memory technology, Henrici distinguishes two general types of storage: non-volatile and volatile storage. Non-volatile storage can be divided into read-only (fixed after manufacturing), Write Once Read Many (WORM) and read-write which set the access privileges to the memory. The opposite of non-volatile storage is called volatile storage and is used for example to perform calculations after power-up. Besides, Henrici mentions tags which are able to check passwords or implement ciphering algorithms to ensure data privacy. To visualize the tag's data and to provide real-time measurement, passive tags can be equipped with displays, buttons and temperature sensors.

To maintain the security and authenticity of each RFID tag, Henrici [1, p.93 ff.] depicts four implementation methods of identification. The first and easiest method is called 'regular identification'. It implies that each tag sends its complete identifier to the reader within a Single Logical Message Exchange (SMLE). Another method is called 'implicit identification'. It uses information that has not been provided explicitly for particular identification purpose. Thirdly, a more sophisticated and secure method to identify tags which is the 'multistep identification' method. As the name of this method is very self-explaining, one should image the next three identification steps: In the first step, only parts of the identification information is revealed. After that, an authentication and authorization step follows. Once, being authenticated and authorized, more identification information will be revealed.

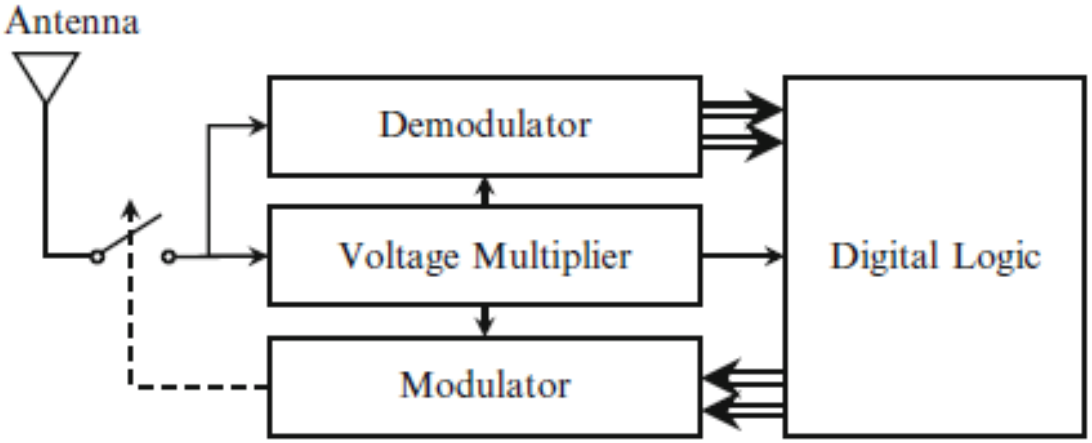


Figure 2.1.- The design of a RFID tag [6, p.13]

Other than the mentioned methods, Henrici describes a very secure method to identify tags which he calls 'encryption and shared key identification'. As an advantage, this identification method protects every information contained in an identifier which can be transmitted in encrypted form. The vast amount of information requires a high internal storage of the tag, like given by active tags. To arrange an encrypted transmission of the information from passive, low cost tags the identifier needs to be calculated outside the tag and then stored on the tag (directly in enciphered form). So, there would be no additional expenditure to enable encrypted and shared key identification.

As above mentioned, RFID tags can be categorized in different ways. In contrast to Tamm and Tribowski 2.1.1.1, Rezaiesarlak et al. [6] describe three different categories [6, p.9 ff.]: Firstly, RFID tags can be distinguished between inductive and radiative (or backscattering-based) tags. Inductive tags are the ones which work below a frequency of 100 MHz and their operating principle is based on inductive coupling between the reader and the tag coil-like antennas. Radiative tags work in the UHF (300 - 3000 MHz) frequency band, providing more reading range as no coupling between reader and dipole-like tag antenna is required. Secondly, there can be differentiated between active and passive tags as mentioned above. Lastly, there are both chipped and chipless tags.

2.1.1.2.- RFID readers

In this section, RFID readers will be explained in detail. To start with, one has to imagine existing objects which are tagged with a RFID tag. To implement functionality to these tags and to connect them to a middleware or a backend system, a detector is needed. This detector is the RFID reader which consists of an antenna, a power supply (for passive tags), a microprocessor (to control devices) and an interface for forwarding data to the processing backend system [1].

Generally, two different types of readers can be distinguished: Stationary and mobile readers. Stationary readers need to be integrated into the existing system architecture by additional middleware [5, p.133 ff.]. Likewise, direct coupling between application systems is not possible because of the amount of data which has to be handled, the lacking ability of being a real-time system and the limited possibilities of RFID readers to produce the required process information. To give an example of the use of stationary readers, they are oftenly used for goods receiving or stock management. Furthermore, stationary readers are fixed to a specific location and need permanent network connection. Additionally, the antenna and reader are spatially separated from each other [2, p.17 ff.].

On the opposite, mobile readers do not need permanent network connection and are used for instance to query prices in a supermarket. They are usually integrated into mobile devices, connected to laptops, Personal Computer (PC)s or tablets. To connect themselves to an existing system, mobile RFID readers need a device driver which enables the communication between reader and the installed application on the device [5, p.133 ff.]. Furthermore, the antenna as well as the reader itself are integrated into their casing [2, p.17 ff.].

Nevertheless, there exists the possibility of conntecting various antennas to one reader to extend the range of field. As mentioned in the section before 2.1.1.1, RFID tags and readers communicate via electromagnetism. The reader's detection range depends on the frequency as well as the electromagnetic field [1]. In general, four frequency ranges can be differenced: Low Frequency (LF) (125-134 kHz),High

Frequency (HF) (13,56 MHz), UHF (868 MHz-915 MHz) and Microwave (2,54 GHz-5,8 GHz). Each frequency range has its own physical characteristics, such as the needed size of antennas or the read range.

According to Vizinex [7], an american company with site in Pennsylvania (U.S), HF tags can be used for short read ranges (up to 3 inches). They are usually tagged to tissue samples, blood and critical fluids. Furthermore, HF tags work well in proximidity to liquids as well as human tissues. UHF tags provide longer read ranges and can be detuned by proximity to tissue, fluids and metals. These tags are typically used to track and locate critical medical devices, manage inventories of medical items and track as well as identify patients. Moreover, UHF tags are compatible with worldwide standards and easily deployed because of the compatibility with widely available and competitively priced RFID readers.

Furthermore, each reader has its own electromagnetic field. Such fields are distinguished into near field and far fields: Near fields, also called magnetic or electric fields work with induction and capacitive coupling whereas far fields consist of electromagnetic waves. The measuring unit of electromagnetic fields is called field strength and the maximal field strength depends on national regulations. These national regulations limit the electromagnetic compatibility to avoid disturbing other systems. The functionality of passive tags within near field is different from passive tags in far field. In near field, the tags send data to the reader using load modulation.

This mechanism does not work in far fields: Here, the sended frequency is backscattered [1]. All in all, readers are able to query tags and to read and write tag data. But the storage of information and the information processing does not take place in readers or tags, but in the middleware or backend systems. These will be explained in the following paragraph.

2.1.1.3.- RFID backend systems

As Henrici [1] mentions, the backend can be divided into two parts: Middleware and applications. Both of them run on the same computer within the same network which

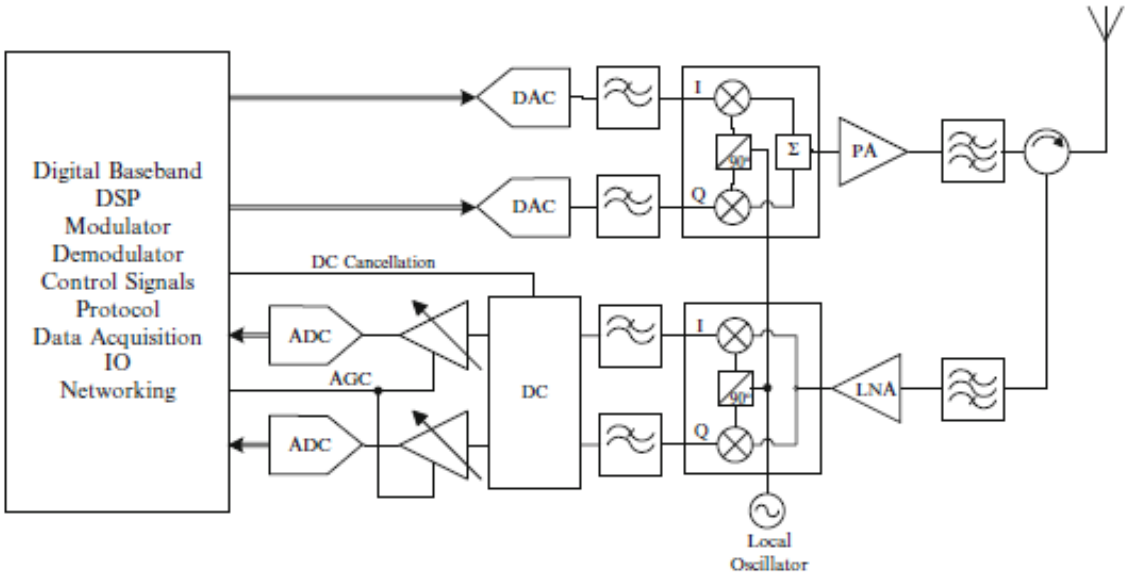


Figure 2.2.- The design of a RFID reader [6, p.17]

is important for the permanent connection to RFID readers and all existing tags. The advantages of a middleware in this use are that no adaption of appications is needed, an open and neutral interface for other applications is provided. Besides, as the middleware is used to aggregate and filter data, the tags only have to identify objects. As a result, modularity of the system is maintained.

2.1.1.3.1 RFID Middleware The RFID middleware can be defined as follows: '[...] the software component for preparation and deployment of RFID data which enables the integration of RFID readers and further infrastructure into the operational application systems [...]' [2, p.20 ff.]. According to this definition, there have to be considered three fundamental functions of the middleware: Reassessment, Aggregation and Tranformation.

To start with, during the reassessment, all received data from the antennas can be redundant or flawed. This redundance is caused by different antennas detecting the same RFID transponder at a time or because one RFID transponder has been detected multiple times during one time frame. Flawed means that transponders were not detected in the intended field or they were detected incorrectly. In second place, aggregation defines the process of summaring all contextual information into one single

RFID information ('together into one'). In third place, the term 'transformation' is used by syntactic and semantic means.

To come to a conclusion, the RFID middleware improves the management of readers by abstracting from technical details. Furthermore, it provides a scalable solution which reduces the unneeded complexity which is transmitted to the users [2, p.20 ff.].

2.1.1.3.2 Data storage concepts Concerning the data management of RFID systems, Tamm and Tribowski suggest three general concepts of storing tag-specific data: 'Data-on-Tag', hybride forms and 'Data-on-Network' [2, p.22 ff.]. 'Data-on-Tag' is a highly recommended method because of the decentral data storage. It improves the user's privacy accessing object-referred information. Moreover, the 'Data-on-Tag' strategy is useful if only relevant and necessary information are contemporarily needed. In addition to that, if the system is not available, the processes can be executed with the tag's information. Furthermore, 'Data-on-Tag' brings many advantages with it like for instance a raised reliability of the entire system because the processes are decoupled from central system components.

On the contrary, 'Data-on-Network' implies a central data storage. Further, it can be easily standardized because only the identification number has to be standardized. Further, the only additional requirement remains the working network connection. Mentioning the different data storage concepts in RFID systems brings up another term which is used to detect e.g. product piracy: Complex Event Processing (CEP). CEP uses compound read events to detect the multiple capture of one identification number (at different places). If multiple captures of the same identification number are found, the mechanism concludes to a copy of the RFID transponder.

Actually, data is barely stored on RFID tags because of the limited resources in low-cost tags. It is recommended [1] to store tag information on an encapsulated database. As an advantage, databases provide high flexibility to change data or to execute queries without the tags being present. Furthermore, the backend infrastructure should use a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocol to ensure a

secure transmission of data. Finally, the data would be transmitted and stored in a backend infrastructure on a central storage [1].

2.1.2.- Functionality of RFID system

When developing an RFID system, it is important to think about the unique identification of each object. To enable a reliable identification of objects, only one RFID tag should be attached to each object. The tag itself has a 'read-only' or in some cases 'rewrite' internal memory which enables users to get or change the object's information [3].

Secondly, the RFID reader generates magnetic fields to enable the RFID system to locate objects (via tags) within its range. Additionally, the high-frequency electromagnetic energy and the query signal which is generated by the reader triggers tags to reply to the query. Each query can have a frequency of 50 times per second [3]. Thus, it is possible to generate large quantities of data which have to be filtered by supply chain industries.

Each filter is routed to a backend information system, using a software similar to 'Savant' which is used to control the data. 'Savant' acts like a buffer between the HIS and the RFID reader [3]. Besides, Tamm and Tribowski [2, p.18 ff.] distinguish between three classifications of RFID systems: 'Close-coupling-systems' ($\leq 1m$ range), 'Remote-coupling-systems' ($\leq 1m$ range) and 'Long-range-systems' (¿1m range).

2.1.3.- Security and Privacy of RFID systems

Security and privacy in the healthcare sector is a very important and highly discussed issue. As these are very large issues, which could not be described within a few paragraphs, there will be depicted some examples of threats. In the second section 'Solutions and Methods against Threats' 2.1.3.2, five important recommended countermeasures will be described. At this point, especially the term of privacy should be defined clearly. According to Tamm and Tribowski [2, p.90 ff.] privacy stands for the right of an individual to keep certain aspects of his life private. Aspects refers to the

informational self-determination which should not be controled by further instances, like e.g. systems or third-parties. Additionally, privacy is considered as basic right which is also defined in the Federal Data Protection Act and includes explicitely the protection of personal data.

2.1.3.1.- Security Problems and Threats

As Henrici [1] mentions, there exist two fundamental fears about the RFID technology. The first fear concerns marketing purposes, such as creating very detailed customer profiles which lead to a vast amount of information. Secondly, the technology offers the possibility to keep people under surveillance which implies advantages and disadvantages. As an advantage, the patients' life gets more confortable and companies will be more productive. As a negative result, people's privacy is violated and the application's security is not addressed properly.

Aside from the two fears, Henrici describes several risks of RFID systems, such as the ease of disrupting the service which indicates data security and privacy problems. When talking about security, one should distinguish between security of systems and services and the security of data and information. The last point can only be ensured by secure systems [1].

In the following, some security and privacy risks using RFID technologies will be explained. To start with, one should think of his passport and the data which is stored on it. The new passports have an internal RFID tag which enables readers nearby the passport to read out all data and to copy them as well. As mentioned in section **??**, passive RFID tags are cheap, do not have their own power supply and can be read through a nearby reader. So, reading out the passport's data would not be very complex. Moreover, Henrici mentions product counterfeiting in pharmaceuticals which can cause a lot of harm, like the death of patient's. Nevertheless, the drug market is bound to strong regulations, like for example through the Federal Drug Administration (FDA). To detect and reduce product counterfeiting, RFID tags need to prove genuineness of original products to patients and should inhibit cloning them.

In his book, Henrici mentions six cases of possible attacks to RFID systems [1, p.31 ff.]. The first attack is called 'Illegitimate reading of data' and describes the possibility of side-channel attacks which use the communication protocol between passive tags and backend systems. As described in section 'RFID backend systems' 2.1.1.3, passive tags are used more often and are less expensive than active tags. Nevertheless, the vulnerability of synchronizing each tag with the backend system through a protocol enables attackers to bypass normal protocols so that they can readout all transmitted data.

The second possible attack, Henrici mentions, is called 'eavesdropping of data'. It is caused by the problem of the public and shared communication channel between readers and tags. Compared to 'illegitimate reading of data', everybody near enough the communication channel is able to eavesdrop the conversation because of the use of passive tags. Particularly the 'forward' channel from reader to tag has a stronger magnetic field than the opposite direction which makes it more easily being eavesdropped than the backed one.

Thirdly, Henrici declares 'cloning or mimicking of tags' as a third threat. His definition of cloning a tag is restricted to creating an exact logical copy of an item which is not distinguishable from the original tag on the protocol level. There might exist some minor differences like the power consumption or time response but the replica cannot be detected with ordinary readers but only with appropriate equipment. The second term 'mimicking' defines the action of infiltrating incorrect data into the RFID system. To show an example, the location might be used for authentication of items. By mimicking a tag, the location can be manipulated and items might appear where they do not exist in reality.

Fourthly, 'recognition of objects' represents another possible threat of RFID systems. In particular, when persons have been detected, they can be used to explore customer habits. Or, in case of patients who wear implants, these might be recognized and the medical information stored on each implant might be abused. In general, each person who carries objects with affixed RFID tags, like wristwatches, shoes etc. might be recognized by an attacker.

Next, the possibility 'tracking of objects' should be considered carefully since tracking of persons can cause many privacy violations. Henrici distinguishes two types of tracking: The first one is called 'direct mapping' and refers to the tracking of RFID wristwatches or glasses. 'Direct mapping' is only possible when the distance between detector and tag is short and the do not exist many tags in one place. Failing that, other items might be tracked by detecting their constellations to each other. These constellations can lead to unwanted creation of movement profiles and the abuse of infrastructure for surveillance by a totalitarian government.

Lastly, Henrici defines the threat of 'causing malfunction' which means that attackers (after having abused one of the above mentioned possibilities) are able to render RFID system malfunctioning. This malfuncionting can be revealed by physical destruction or chemical treatment of tags.

2.1.3.2.- Solutions and Methods against Threats

First of all, data security should always be maintained by the RFID system. But what are the exact countermeasures to prevent an attack on an existing RFID system? When Henrici [1, p.64 ff.] talks about solutions and methods against security threats, he calls them 'Goals of Security and Privacy'. In his book, these goals refer to the possible attacks or threats mentioned in section 'Security Problems and Threats' 2.1.3.1. In the following, the countermeasures will be explained.

'Illegitimate reading of data' can be prevented by controlling data access and ensuring data integrity in RFID systems. False data should be infiltrated because of illegitimate access. 'Eavesdropping of data' can be coped with implementing means for detection and recovering so that the system should keep running even if attackers try to put it out of service. Besides, the integrity of system should always be kept. Another strategy preventing eavesdropping is to maintain data security. Henrici defines a 'good' RFID system to be able to cope with illegitimate reading of data and to treat all the data confidentially.

'Cloning or mimicking of tags' which can be compared to counterfeiting can be prevented by using authenticity mechanisms to identify specific tags. Therefore, RFID tags that can prove their own authenticity should be preferred.

Unwanted 'recognition of objects' can be avoided by developing technical models that provide suitable trade-off of functions. If a function is not wanted by the user, e.g. to allow everyone in the surroundings to read out all RFID attached object, he can adjust this by defining different user roles and rights.

Regarding the realization of the above mentioned goals, there exist many challenges which have to be faced. Henrici [1, p.66 ff.] describes four general challenges which will be explained in the following paragraph.

First of all, since there are different parties, like e.g. logistic companies and customers which have different needs, the developer has to meet all of their requirements. For instance, the different user needs might be realized by developing different views which depend on the particular user role.

Secondly, developing a secure RFID system is a multidisciplinary challenge [1] including six different departments: Computer science (designing communication protocols and the middleware), electrical engineering (realizing the required functionality in hardware and physical layer of communication between tags and readers), mathematics (developing basic cryptographic primitives and theory of probabilities for different areas), economics (adapting the application's constraints imposed by laws of market and assessing real world applicability of approaches), social sciences (including user's requirements, such as privacy and usability) and law (maintaining a legislative basis among people and organizations).

Thirdly, there are more requirements to be faced than 'only' security and privacy, such as low costs or coping with few capabilities and resources. Besides, the enrollment of an RFID system, e.g. in a hospital with many distinctive departments, leads to an inter-organizational operation. To implement this inter-organizational operation, several standards have to be integrated.

Last but not least, additional requirements have to be considered: Scalability of the system, dependability, low complexity of system, robustness, transparency and usability etc. Henrici claims, that the safeguards should not limit the read range and the speed of reading. Moreover, when using cryptographic primitives, migration paths should be considered.

2.2.- NoSQL technology

2.2.1.- Characteristics of NoSQL Databases

There are many possibilities to store data from application systems. Structured Query Language (SQL) can be seen as one of the fundamental database technologies used since the 1980/1990ies [8, p.137 ff.]. The technology of SQL offers advantages, such as consistency, security and integrity of data, as well as the protection of transactions. On the other side, there come along many disadvantages while using SQL.

To give an example, checking the integrity of data in case of a higher amount of data implicates the need of a higher processing power. Furthermore, facing large-scale development, the efficiency and performance of SQL based systems are decreasing. Moreover, the flexibility and data handling demonstrates another challenge when using SQL. Actually, in practice, the performance is often more important than the consistency, e.g. in social media.

In the past ten years, there were some NoSQL database technologies coming up. In 2000, these databases were called 'Web-Scale-Databases' because of the need of data storage systems which should handle with the large amount of data of web services [8, p.221 ff.].

The term NoSQL can be understood as 'Not only SQL' which signifies extension of the existing SQL functionalities. Some data specialists prefer the context of 'no relational databases' which is controversial because of the use of graph databases which deal with relations between nodes.

In the next section, some examples for the NoSQL data management will be explained and compared to the usual SQL data management. According to Meier and Kaufmann [8, p.3 ff.], SQL databases are formed on a relation-based model which contains tables with entries. Furthermore, each entry has attributes which are defined by a validating range. A table is defined by its table name, the attribute's name and an identification key. It can have both a column or row order or none of these. The relational model indicates that every table is a set of random tuples and that the relations between data is realized by using tables. The result of SQL queries are always tables which can be uniquely, minimally identified by their identification or primary key.

In contrast to that, the technology of NoSQL offers more possibilities to store data, like for example in key-value stores, column stores, document stores or graph databases [8, p.16 ff.]. These four types of NoSQL databases are also called 'Core-NoSQL-Models'. Besides, other NoSQL database models like object databases, Extensible Markup Language (XML) databases or grid databases are defined as 'Soft NoSQL Models'.

To give some examples of the 'Core-NoSQL-Models', their fundamental characteristics will be described in the following. The key-value stores (e.g. Cassandra) provide the simplest way to store data by using an identification key and a list of values. Document stores (like e.g. MongoDB 3.2.2) store the data in form of structured text data like JavaScript Object Notation (JSON) or XML. In contrast to key-value stores, document stores have a pre-defined structure but are schema-free which means that the data structure can be changed over the time.

Graph databases (e.g. Neo4J) introduce a new way to store data: It introduces a graph-based model. Each graph consists of nodes and edges which connect the edges and demonstrate their relations. Each node can have concepts and object. Both, nodes and edges have a label and can contain properties. Each property contains an attribute and a value. The query language for graph databases is called 'Cypher' [8, p.16 ff.] which is a declarative language. Users of Neo4J and other graph databases are able to specify their retrieval query by defining nodes and edges. By evaluating all possible

paths (or connections between nodes and edges), the database system calculates all requested patterns.

Generally, NoSQL technologies are popular for their high availability as well as their protection against system failures by using different replication concepts, e.g. 'Consistent Hashing' [8, p.11 ff.]. In addition, NoSQL is characterized by its vertical and horizontal aligned scalability, its weak or non-existent restrictions concerning schemas and data models. Along, NoSQL databases offers simple data replication and easy access through API [8, p.221 ff.].

Edward and Sabharwal discuss the pro and cons of NoSQL technologies in their book [9, p.17 ff.] which will be given in the next paragraph. On the one hand, NoSQL offers high scalability, manageability and administration (such as automated repairs, distributed data etc.), low cost and flexible data models. But on the other hand, using NoSQL factors like maturity, limited query capabilities, administration (installing and maintaining solutions) and limited expertise (developer and administrator community are limited).

A Database Management System (DBMS) defines a software which describes, stores, queries data independently from an application [8, p.2 ff.]. It consists both of a storing and managing component. The storing component is composed of all data which has to be stored in organizational form and their description. The managing component contains a querying or manipulating language to evaluate data and to change them, such as access control units and an user interface. When it comes to the use of web applications and a heterogeneous data pool in real-time, SQL databases are often not suitable for these problems. In this case, a NoSQL database should be considered.

2.2.2.- Excursus: BIG DATA

The term 'Big Data' has been emerged during the past 10 years. Due to the enormous data pool, e.g. in social networks or user analysis, which is not easy to manage with usual software tools, new data technologies were needed. As a solution, NoSQL

technologies have been arised. Furthermore, Big Data refers to unstructured data which comes from different sources [8].

Edward and Sabharwal define Big Data as the following: It is a '[...] term used to describe data that has massive volume, comes in a variety of structures and is generated at high velocity. This kind of data poses challenges to the tradtitional Relational Database Management System (RDBMS) used for storing and processing data. Big Data is paving way for newer approaches of processing and storing data.[...]' [9, p.1 ff.]. Furthermore, Edward and Sabharwal refer to an explosion of data created by smartphones, social networking sites, in short words from various sources in various formats, such as video, text, speech, log files and images. The type of stored data varies by its 'sector', e.g. retail and whole sale, administrative parts of government, financial services mainly generate text or numerical data (including customer data and transaction information).

On the other hand, in the healthcare, manufacturing, media and communication sector especially multimedia as well as image data is produced. To give an example, X-Rays, CT and other scans dominate the storage volumes in healthcare.

Another important point, when it comes to large amounts of data is the consumation model or the various sources from which data is produced and subsequently consumed [9, p.6 ff.]. Edward and Sabharwal declare two models: In the old model, few companies produced data and all others (users, clients, etc.) consumed them. But nowadays, they claim that '[...] all of us produce data and all of us consume them [...]' [9, p.6 ff.]. This creates a new challenge of developing multiple user data models and make the existing data management system more scalable than in the past.

2.2.2.1.- Velocity, Variety, Volume

When talking about Big Data, often there come up three descriptive terms: Velocity, Variety and Volume.

Velocity refers to the real-time high-speed evaluation of the upcoming data [8], also known as 'real-time insight' [9]. Edward and Sabharwal describe velocity as 'Data in

motion' [9, p.7 ff.]. Besides, they mention that if data cannot be processed at required speed, it losses its significance. For many companies and organizations, it is very important to process data both when it is moving as well as when it is static.

Variety means that there are distinct formats of data: structured (e.g. integer, string), semi-structured and unstructured data [8]. Edward and Sabharwal describe variety as 'Data in many forms' [9, p.7 ff.]. As mentioned in the last paragraph, data can vary from simple text files, log files, streaming videos, photos, meter readings, stock ticker data, PDFs and many other unstructured format.

The last characteristic of Big Data, volume, deals with the high amount of data [8]. Edward and Sabharwal describe volume as 'Data in many forms' [9, p.7 ff.]. They see a reason for the higher volume in businesses becoming more transaction-oriented and the number of transactions is increasing. Moreover, more devices are connected to the internet which increases the size of data.

Calolas et al. refer to challenges like e.g. dealing with tremendious amounts of data, unstructured data (diversity of Online Social Networks (OSN)) as well as the complexity to challenge analyzing social networks data [10].

2.2.2.2.- Usage of Big Data

Edward and Sabharwal depict five large use cases of Big Data which will be explained in the following [9, p.9 ff.]. Firstly, visibility of data is very important for many companies. For example if data is accessible across departments of a company, it can be readily integrated. This reduces the search and processing time, improves product quality according to present needs [9, p.9 ff.].

Secondly, discover and analysis information is another use case for large amounts of data. After capturing detailed data, e.g. of inventories, employees or customers, new information or patterns will be discovered and analyzed. The captured information and knowledge can be used to improve processes and performance of companies [9, p.9 ff.].

Thirdly, segmentation and customizations can be effected by means of using Big Data. To give an example, the segmentation of customers is based on various parameters and can aid in targeted marketing campaigns or tailoring of products to suit the needs of customers [9, p.9 ff.]. Fourthly, Big Data can aid, improve or automize decision making by using Big Data analytics which can minimize risks and uncover valuable insights. Lastly, Big Data can strengthen innovations in existing products by using data gathered for actual products [9, p.9 ff.].

2.2.2.3.- Big Data challenges

The issue 'Big Data' not only offers many possibilities or use cases, but also faces many challenges [9, p.11 ff.] which will be discussed in the consecutive paragraph. To start with, Edward and Sabharwal indicate policies and procedures which can constrain the use of Big Data. They refer to data privacy, security, intellectual property of organizations. Furthermore, in order to comply with various statutory and legal requirements, Big Data has to face the challenge of data handling, which includes issues around ownership and liabilities around data. In second place, the access to data has to be controlled precisely.

Since some data might be available to third parties, the gaining access poses a legal, contractual challenge. After that, in order to handle Big Data, new tools as well as technologies have to be built specifically. Moreover, there might exist legacy systems in several organizations or companies which have to deal with Big Data. Besides, there exists a lack of experienced resources in these newer technologies which is also a challenge to.

Most of the legacy systems are designed to work with structured data [9, p.11 ff.]. Since legacy systems are created to perform fast queries and analysis on tables and columns, they cannot be used to hold or process Big Data (which contains unstructured data). Another important point is the data storage for Big Data. Currently, in many companies, data is stored on big servers (using Network Attached Storage (NAS) or Storage Area Network (SAN) systems) [9, p.11 ff.]. With the increasing data, the server size and backend storage size has to be increased.

Lastly, when handling with Big Data, data processing is very important . Usual algorithms in legacy systems are designed to work with structured data and are limited by data size. Therefore, legacy systems are not capable of handling processing of unstructured data, high volumes of data, speed etc. To capture value from Big Data, the deployment of newer technologies are needed.

2.2.2.4.- Big Data technologies

Edward and Sabharwal propose several ways of implementing Big Data technologies [9, p.12 ff.] which will be depicted in this paragraph. To begin with, Big Data needs new storage and processing technologies which are designed for large, unstructured data. After that, other technologies like parallel processing, clustering are emerging in order to handle Big Data. Additionally, large grid environments, high connectivity and high throughput offer new fields for software developer and data scientists. Finally, cloud computing and scale-out architectures have been arised during the last years.

2.2.3.- Example for using NoSQL Databases: 'Socii System'

To give an example of the use of NoSQL technologies, the next section will focus on explaining a developed system 'Socii'. 'Socii' was developed from Jroge Daniel Calolas, Alda Lopes Gancarski and Pedro Rangel Henriques. In their article 'Online Social Network Analysis Visualization Using Socii' [10, p.218-228], they describe the several technological challenges they faced during development but also the benefit of this social network analysis and its scientific importance. In short words, 'Socii' is a system which enables the analysis and visualisation of social networks by helping OSN users to exploit and understand their own networks through a user friendly interface. During development, Calolas et al. were facing four main principles: simplicity, accessibility, OSN integration and contextual analysis.

One big problem before developing of 'Socii', Calolas et al. mention was the observation of social structures and the analysis of these networks. By establishing 'Socii', the authors Calolas et al. wanted to proprose to fill the gap or struggle that OSN users have in understanding their network. To give an example of these struggles,

there were three main topics, Calolas et al. were facing: 'how relationships evolve along the time', 'what role play these friendships within the network' and 'how they can analyze and visualize their networks based on social properties (such as mutual relationships, geographical positions, personal tastes and preferences or hobbies)' [10].

2.2.3.1.- Structure and flow of data analysis and visualization systems

When introducing 'Socii' as an social analysis tool, Calolas et al. also talk about several tasks and steps during the process of data analysis and visualization systems. First, data has to be extracted (through APIs, web crawlers and web scrapper. Second, data is achieved which requires careful selection of relevant data to store [10]. In order to have an efficient system that provides good structure for data analysis, one needs to select the data carefully. In the third step, data is explored by defining of what one user wants to do with the data. Furthermore, during this step, there are two important questions: What are the applications that can be seen for the stored data? How can the system digest and transform data in order to make it useful and interesting for the end user?

Lastly, in the fourth step, data is visualized which means the kind of presentating or showing the transformed data is chosen by the user. Particularly, the work of the data scientist has a huge impact on the end user. In generals, the data visualization targets a general audience. By extension, 'Socii' affirms web availability, OSN's integration, contextual analysis and being a trade-off for such gains the system performance.

2.2.3.2.- Main functionalities of 'Socii'

One of the main functionalities of 'Socii' is information extraction and data mining [10, p.223]. These include extracting some user networks form given OSNs by calling web crawler modules which then return the extracted information. After that, an extraction manager sends the extracted data through a simple data mining process in order to data normalized before it is stored in the database (MongoDB). Web crawlers are implemented in Python and the crawling operations are performed using XPath selectors to extract the information that are needed to build the network.

Generally, the main functionalities of 'Socii' amount to OSN contextual netowrk analysis (with relatively low complexity) [10, p.227]. Moreover, 'Socii' intends to integrate data from different OSNs by processing a set of node properties displayed together with the network structure. Roughly, Calolas et al. implemented three features of 'Socii'. To start with, 'configurable, parameterized analysis' enables users to select several metrics upon a given network.

Consequently, 'clear, intuitive social graph visualization and interaction' refers to the visual web component of 'Socii' which provides users a set of visual features (coloring, node discovery etc.). Thirdly, 'Socii' offers an 'organized overview upon SNAs and OSN data' which include visual components that aggregate SNAs metrics and OSNs information. Thus, users are able to cross information from both and can derive conclusions from intersecting the information [10].

2.2.3.3.- Limitations of 'Socii'

As shown in the last paragraph, 'Socii' offers many features to analyse Big Data especially in social networks. Nevertheless, Calolas et al. depict some limitations and disadvantages which will be explained in the following. There exists a technical and architectural struggle of feeding the system through an extraction pipeline built on top of the web crawlers. This is known and proved by a very slow, limited and error-prone method for data extraction and the authors call it the 'bottleneck' of their built system [10, p.227].

2.2.3.4.- Future outlook of 'Socii'

According to Calolas et al., 'Socii's implementation should be completed to work with other OSN's. Furthermore, it can be extended to perform evaluation and validation of system with users having accounts in several OSNs and several profiles.

3. Development of RFID application for management and tracking assets and drugs

The fourth and last chapter of this Bachelor Thesis gives information about the development process of a RFID tracking application. In the following, the system's requirements 3.1 are described. Additionally, all used platforms and technologies are mentioned in section 3.2. Then, the specific development process 3.3 is depicted which demonstrates the software's architecture introducing some diagrams. Lastly, several planned tests as well as their test results are discussed in section 3.4.

3.1.- Proposed solution of application

3.1.1.- Aim of developed application

The aim of the proposed system is to improve logistics and the management of drugs in hospitals. Considering the risk of errors or mistakes that might occur when administering drugs to patients, the developed mobile application can prevent these situations. Furthermore, when patients are in a critical situation, the system offers the possibility to respond faster to emergencies as well as to improve patients' monitoring. Finally, the placement and usage of medical assets can be pursued and controlled better.

The above mentioned reasons lead to the necessity of disposable, low cost RFID tags which might be tagged to patients, drugs and medical assets. Moreover, there has to be established a network between the mobile devices, the RFID reader and the synchronizing software.

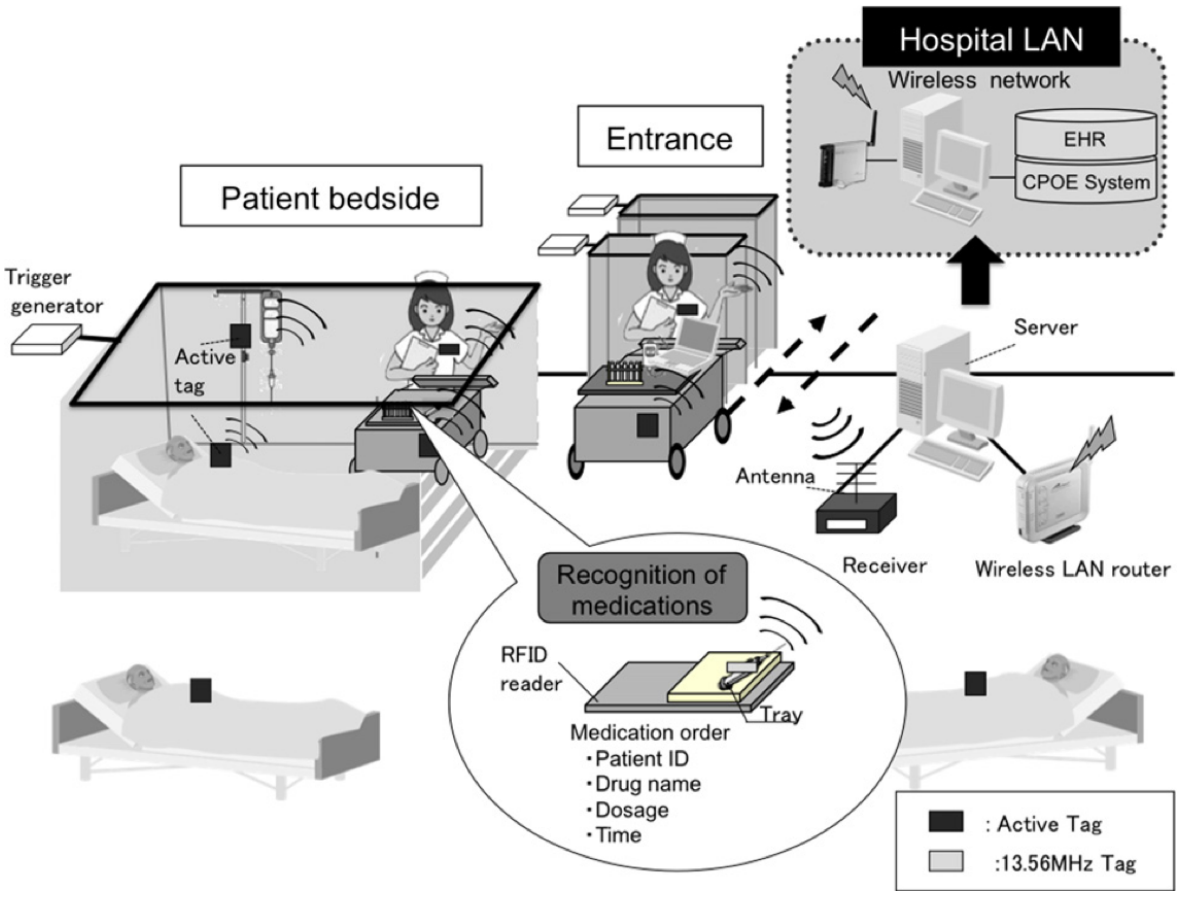


Figure 3.1.- Existing RFID systems, based on 13.56 MHz short-range, inductive RFID tags

3.1.2.- Scope of developed application

In the past, the existing RFID systems for management and logistics in hospitals were mostly based on 13.56 MHz short-range, inductive RFID tags. Figure 3.1 gives an example of an existing RFID solution for management in hospitals. The given system has several advantages as well as disadvantages. On the one hand, as an advantage, using 13.56 MHz short-range, inductive RFID tags is a mature technology which is reliable. Further, the past solutions imply low-cost RFID tags which are easy to use. On the other hand, the disadvantages of the mentioned solutions are their complex infrastructure and the short-range.

Therefore, the present bachelor thesis introduces the use of 868 MHz frequency band while using a similar RFID architecture as the ones already tested and deployed. As an advantage of using 868 MHz frequency band, the proposed solution implies a longer range, wider coverage and the required hardware can be simplified. Further, the proposed solution of this bachelor thesis provides a mobile application for users which can be run on tablets and smartphones.

In a nutshell, this bachelor thesis modifies the Radio Frequency (RF) hardware as well as the front end whereas the back-end and core network remains the same.

3.1.3.- First concept of application

Figure 3.2 describes the very first draft of the workflow of the application. Actually, it is not very precise but should help to describe the general scope of the medication tracking system. At the top of the picture, there is an Excel program which stores all data of the database and converts them into *.CSV files. These converted files should be imported into the running MongoDB instance on the PC.

On the same computer is running Matlab which is connected with the RFID reader. All along, MongoDB and Matlab are synchronizing each other. Beneath the PC, the figure shows the RFID reader which is connected to four antennas. At the bottom of figure 3.2, there are drawn four zoned antennas. Each zone can be seen as a

different area or room and can represent a different state, e.g. 'entering' or 'leaving' the stockroom. In the bottom left corner, there can be depicted RFID tags which pass one antenna after another. Each time, an antenna recognizes a tag, it sends the transmitted information to the reader which transfers the data to Matlab.

This first concept was changed due to the use of the 'MongoDB' plugin, which offers Matlab [11]. This is much easier than exporting all received data from Matlab in *.CSV format and then including it into MongoDB as *.CSV data. Furthermore, it supports developing a very fast and real-time application which enables a higher data integrity because of the direct connection to the database (see also figure **??**.

3.2.- Used platforms and technologies

This section focusses on explaining the used technologies and frameworks for developping the mobile RFID application. In the following, the framework 'Nativescript' which can be used for native mobile development will be explained. After that, section 3.2.2 discusses MongoDB, a document store. Finally, the Speedway Revolution RFID reader is presented. In the last section 3.3 of this chapter the challenges as well as the user scenarios are shown.

3.2.1.- Native Development with NativeScript

There exist several ways to create a mobile application, like for example with Android Studio or iOS. But the challenge is to develop a hybrid solution for both systems at the same time. To face that challenge, Nativescript has been established as an open-source project from Telerik in the last years [12].

The free and open source technology enables developers to easily build cross-platform native apps with either Javascript, Nativescript or by using Angular [12]. Regarding its design philosophy, Nativescript was designed to be approachable to developers from various backgrounds [12]. Moreover, it was designed to be both performant and giving access to native APIs, such as Android or iOS.

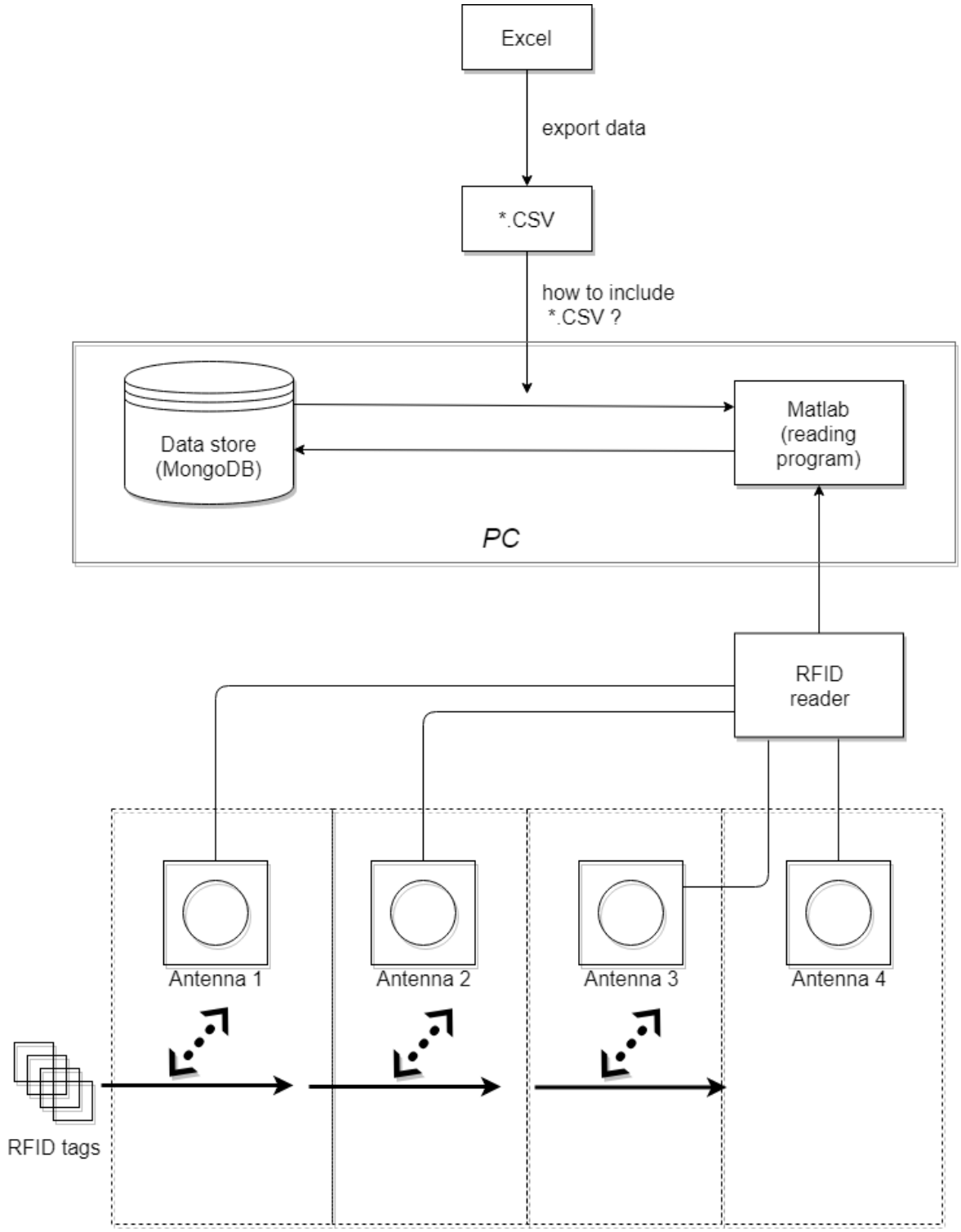


Figure 3.2.- The first draft of the developed system

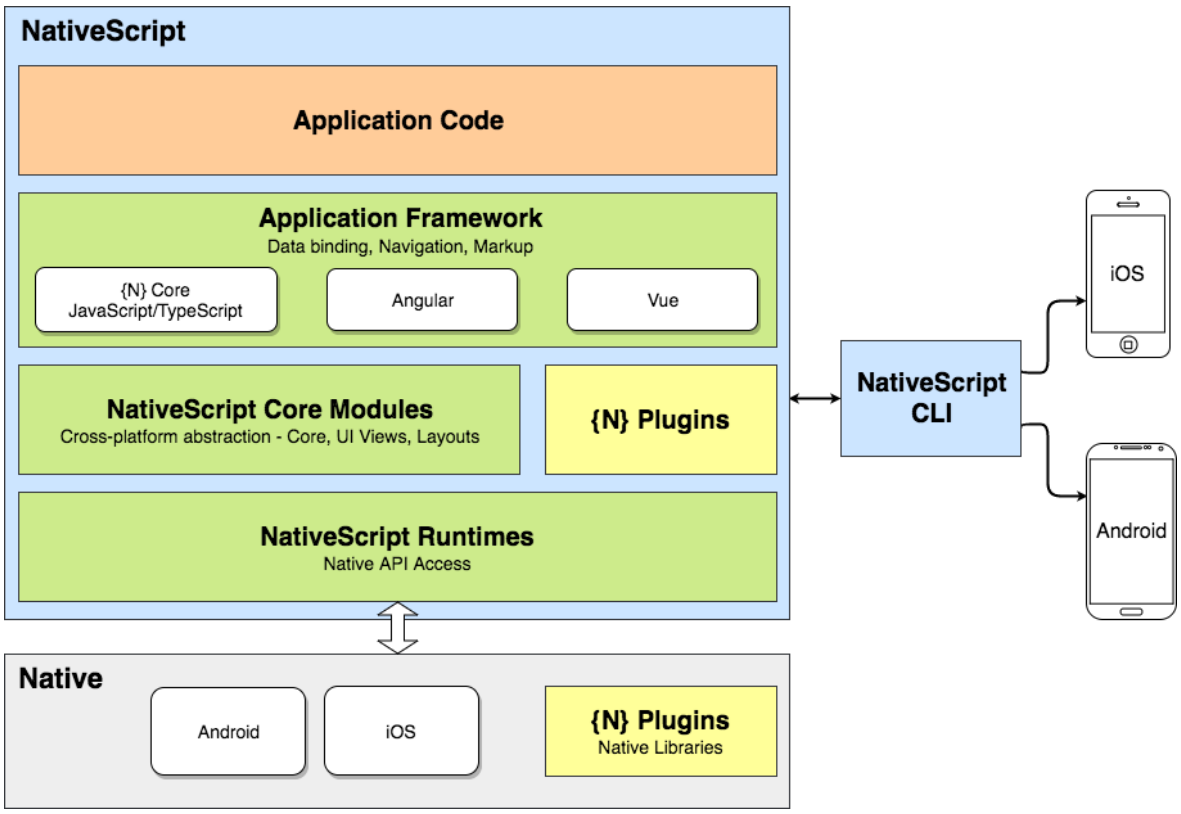


Figure 3.3.- The architecture of NativeScript Applications, adopted from [13]

Figure 3.3 gives an impression of the general architecture of Nativescript applications. When developing such applications, one of the given frameworks can be used (e.g. '{N} Core', Angular or Vue). Additionally, several Nativescript plugins can be selected. Below the 'NativeScript Core Modules', there are located the 'NativeScript Runtimes' which have direct access to the Native system. By running several commands on the NativeScript Command Line Interface (CLI), the developed Nativescript application can be executed on any physically connected device as well as on the installed emulator or in the cloud.

NativeScript applications can also be developed, built and run on the 'NativeScript Playground' [14] which enables independent development. Moreover, NativeScript Playground is easier to handle because the application is executed in the Cloud. Generally, by offering a user-friendly surface, NativeScript Playground is appropriate for beginners who start developing native mobile applications.

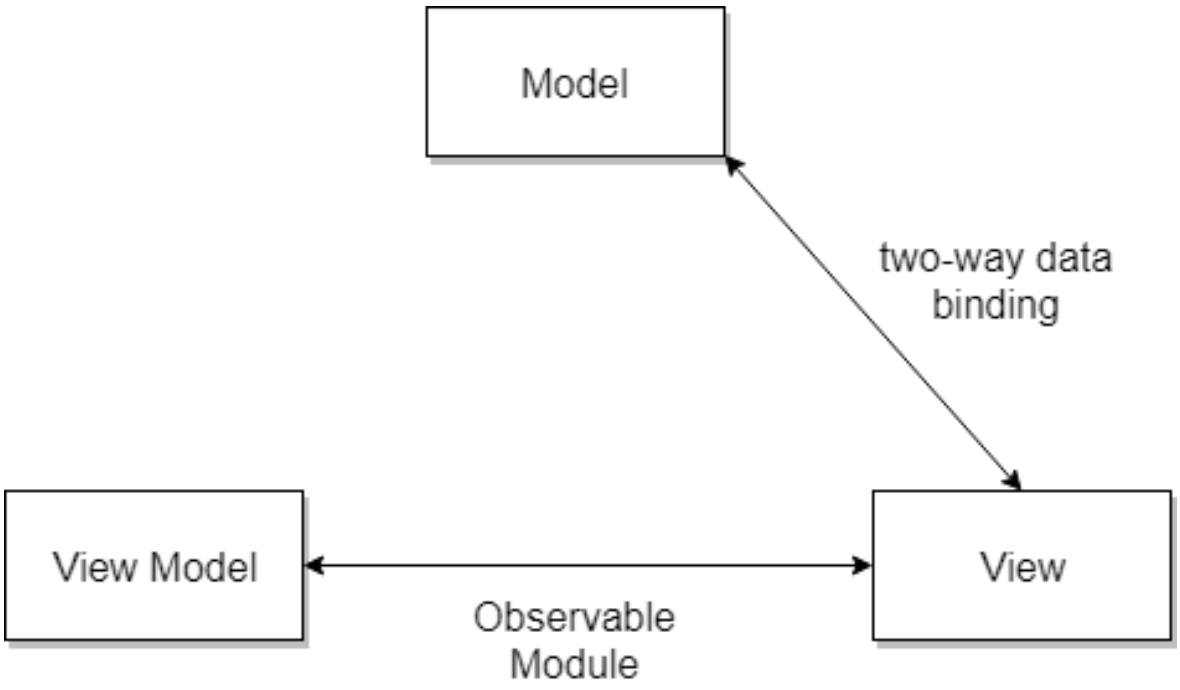


Figure 3.4.- The MVVM application logic, adopted from [12]

3.2.1.1.- NativeScript Application Logic

NativeScript has a Model View View Model (MVVM) application logic (see figure 3.4). In contrast to the popular Model View Controller (MVC) application model, NativeScript offers two-way data binding by using the 'Viewmodel' [12]. In every NativeScript application, the model defines and represents data. After that, the data are bound to the view which represents them in a XML file. The 'ViewModel' contains the application logic and exposes all data to the view.

The data between model and view can either be synchronized as one-way (default setting, the target property updates when a change in the source property occurs) or two-way data binding (all changes in the directions target-source and source-target will be transmitted). To enable the two-way data binding (see figure 3.4), the NativeScript Observable Module has to be implemented.

According to the NativeScript Documentation [12], the model files are called 'Code Behind' because they have the same name as the view file and are written in Javascript

or Typescript. By adding attributes to any XML element in the view file, methods can be implemented in the related model file (in Javascript or TypeScript).

3.2.1.2.- NativeScript Sidekick

NativeScript Sidekick is a solution to run the developed application on unsupported platforms in the cloud. It uses both the local build infrastructure and the cloud build service. NativeScript Sidekick offers users to develop with the provided starter templates, to use verified plugins and to build the app in the cloud. Furthermore, NativeScript Sidekick enables developers to debug, test and refactor their application. To search for plugins and to manage these, developers can use the Nativescript Marketplace [15].

3.2.2.- NoSQL Technology: MongoDB

As mentioned in section Document Store ţefKap2, this type of NoSQL database provides high availability, scalability and partitioning options [9, p.25 ff.]. Nevertheless, there are some disavantages when using MongoDB or any other document stores: For instance, both consistency and transactions are not supported.

In contrast to relational databases, MongoDB does not consist of tables and rows, but of collections containing documents which make it both flexible and scalable [9, p.25 ff.]. Collections can be compared to tables in SQL, but are schemaless. Instead of having one unique schema within the same collection, every document can have its own set of fields, and common fields can store different values across documents.

All data is stored in Binary Structured Object Notation (BSON) documents which assures that related data is placed all together in one place. BSON documents are JSON documents in binary-encoded format. It is the extended form of the JSON data model and is fast, high traversible and lightweight [9, p.31 ff.]. Moreover, JSON/BSON documents contain schema-less models. Each document stores data as key-value pairs, where the value can be left blank. Thus, users of MongoDB have to ensure and check the consistency of their data when adding new data.

One characteristic of MongoDB is the _ ID (key) which can be compared to the label or name of a column in RDBMS. If not explicitely specified by the user, a unique value is automatically generated and assigned to it by MongoDB. Basically, the key value is immutable and can be of any data type except arrays [9, p.31 ff.].

Queries in a MongoDB database use the keys (_ ID) in documents which makes it possible to query documents spread across multiple servers. MongoDB uses primary-secondary replication, where the primary replication accepts the write requests. To be more precisely, if the write performance needs to be improved, the mechanism of sharding can be used.

Sharding means that data will be split across multiple machines which are enabled to update different parts of datasets [9, p.25 ff.]. Besides, this mechanism is automatic in MongoDB, so that as more machines are added, the data is distributed automatically.

3.2.2.1.- Limitations and possibilities of MongoDB

There are many advantages and disadvantages, limitations and possibilities when using MongoDB. The following paragraph will discuss these and give some examples.

Generally, MongoDB offers many features which MySQL does not. To give an example, MongoDB supports secondary indexes, atomic updates at a per document level. Additionally, queries can be executed by using query documents. Furthermore, MongoDB provides replica sets which are based on master-slave replication with automated failover. After that, MongoDB provides a built-in horizontal scaling. Finally, MongoDB can be run everywhere (e.g. on Cloud, Virtual Machine (VM), servers etc.) because it is written in C++.

Another feature of MongoDB are 'Capped Collections' which store documents in the inserted order. When the capped collection reaches its storage limit, documents will be deleted from the collection in the inserted order (analogous to first in first out (FIFO) principle). Capped collections are often used for log files in order to get these automatically truncated after a certain size. In the end, capped collections guarantee preservation order data in the insertion order [9, p.31 ff.].

On the other hand, in contrast to relational databases like MySQL, MongoDB does neither support JOINs nor fully generalized transactions [9, p.25 ff.]. Secondly, when using MMAPv1 as storage engine, the used space is too large because its data directory files are larger than the database's actual data [9, p.226 ff.]. For that reason, it is recommended to use MongoDB's WiredTiger storage engine which compresses all files and reduces the storage size by 50%. Moreover, once a collection is dropped, diskspace is automatically reclaimed (unlike MMAPv1 engine).

Thirdly, when using MongoDB BSON documents, their usage is limited to the size, nested depth and field limits of the specific document [9, p.228 ff.]. After that, namespaces as well as indexes are limited, for instance the maximum size of indexed items has to be 1024 bytes. The number of indexes per collection must not exceed 64 indexes. Moreover, the usage of sharding is limited [9, p.230 ff.]. Therefore, if shards were implemented too late, a slowdown of servers could be caused because splitting and migration of chunks takes time and resources. Thus, Edward and Sabharwal recommend to shard a collection before reaching 256 GigaByte (GB).

Next, when using MongoDB, there exist some security limitations because the database does not provide authentication by default. This enables every user which is connected to the database server, to read, change, add and delete data. In addition to that, the connections to and from MongoDB are not encrypted by default. Therefore, when starting the database server on a public network, it is recommended to use encrypted communications. Therefore, Edward and Sabharwal propose the SSL-supported build of MongoDB (which is available as 64-bit version) [9, p.230 ff.].

What is more, using MongoDB implicates write and read limitations, such as case-sensitive queries and type-sensitive fields since there is no enforced schema [9, p.231 ff.]. For instance, users have to ensure the correctly used type when adding new data. By the same, replica sets which can be used to ensure data redundancy, are limited by the number of the members in every set. When using such replica set, one member acts as a primary member whereas the rest acts like secondary members (a node needs the majority of votes to become primary).

All in all, MongoDB provides many possibilities but also limitations. Generally, it should be clarified that MongoDB is neither adequate to be used in a highly transactional system nor in business intelligence applications where issue-specific databases shall generate highly optimized queries. Finally, it should not be used in applications requiring complex SQL queries. Instead, it is recommended to store high amounts of data and to ensure its availability at any time.

3.2.2.2.- MongoDB: Best Practices

Edward and Sabharwal suggest some 'best practices' when using MongoDB [9, p.234 ff.] which will be explained briefly in this paragraph. First of all, the correct hardware should be chosen. For example, the more internal memory is given, the better the performance of the MongoDB application. Another important point is the CPU: for its productivity, MongoDB needs a fast Central Processing Unit (CPU) clock speed, as well as a high Random Access Memory (RAM). Moreover, using SATA SSD and Peripheral Component Interconnect (PCI) ensures good price as well as performance results.

Concerning the best practices in coding, Edward and Sabharwal propose to set a correct data model. Further, they recommend avoiding application patterns that lead to unbounded growth of document size as well as to design documents for future which are better to handle with MongoDB's drawback. Besides, documents should be created with an anticipated size where ever applicable. But the most important point when coding is to check data consistency [9, p.234 ff.].

With respect to the data safety and how to provide it in the deployed MongoDB database, Edward and Sabharwal both replication and journaling of data [9, p.234 ff.]. Furthermore, the repair should be the last resort for recovering data in case of a server crash. Basically, they recommend to always specify a timeout with assuming the command and to run the MongoDB server instance in trusted environment with access control.

Likewise, concerning the administration of MongoDB, Edward and Sabharwal submit to perform instant-in-time backups of durable servers and to use repair in order to compact databases [9, p.234 ff.]. Correspondingly, managing the replication log is of administrative concern [9, p.240 ff.].

Respecting sharding, it is important to select a good shard key. Moreover, Edward and Sabharwal recommend using three config servers in production deployments to provide redundancy [9, p.240 ff.].

Finally, Edward and Sabharwal introduce 'monitoring' which means that every MongoDB system should be proactively monitored to detect unusual behaviours [9, p.240 ff.]. To put this into practice, MongoDB's free hosted monitoring service, MongoDB Cloud Manager (which contains a dashboard view of the entire cluster metrics, can be used. Besides, MongoDB itself provides 'mongostat' and 'mongotop' to gain insight into the performance of the database instance (e.g. operation counters, active working sets etc. are shown).

3.2.3.- Speedway Revolution RFID reader

The developed system was tested and run on a Speedway R420 RAIN RFID READER [16] with four antenna ports (extendable to 32 antennas with an Impinj Antenna Hub [16]). According to the manufacturer's web page [16], the Speedway R420 RAIN RFID READER can be connected to Impinj Autopilot technology which automatically optimizes the reader's operation for its environment. A test software can be downloaded from the Impinj web page [16].

3.2.4.- Matlab

According to Dr. Knuth's 'Art of Computer Programming', Matlab is an interpreted language, usually used without an compilation step [17, lesson 1]. Actually, the Matlab code is converted into machine code by an interpreter as the program is executed. Together with Java, Matlab counts to the high-level languages which on the one hand enables a faster code development but on the other hand limits the speed of execution.

Matlab provides a large number of functions for common complicated operations in scientific computing (including engineering and analysis) and technical calculation.

Matlab provides important terminology, like e.g. variables (which contain three fundamental types: characters, real numbers, complex numbers), arguments, commands, command line, scripts, functions as well as comment [17, lesson 1]. One of the most common features of Matlab is its 'Debugger' [17, lesson 8] which is very useful for detecting unobvious errors.

3.3.- Application development

3.3.1.- Progress of development

The development process can be divided into four steps which will be described in the following section.

To begin with, during the very first weeks one work package was learning how to create a NativeScript application from scratch. This phase included making experience with the common NativeScript templates, views and framework logic as well as the data binding and data handling.

As a second work package, the challenge of how to connect the NativeScript application to the backend (MongoDB) was faced. As a first solution, the connection to the MongoDB server was implemented using HTTP request/response. But in reality, there might be at least two devices (used by a nurse or doctor) which have to be connected to the database. For that reason, a socket connection (using the Socket.IO library) was established. With this technology, it was possible to connect multiple clients to the server instance and to synchronize all data in real-time.

After that, the third work package introduced the handling of Matlab, its connection to MongoDB and to the Impinj RFID reader. In order to connect to the RFID reader, the Impinj Octane SDK was used. Thus, it was possible to configure several parameters for each antenna of the reader. This enables a precise identification of each antenna as well as the precise location of each detected item or drug. To give an example,

antenna 1 could be set as hallway or floor whereas antenna 2 stands for patient's room. Concerning the connection between RFID reader and MongoDB, there exists a MongoDB plugin for Matlab. This provides several methods to connect to a MongoDB server and to retrieve and change the data from a specific database. Concluding these two implemented functionalities, Matlab serves both as a detection and information medium.

The last large work package refers to the challenge of establishing a stable connection between all components (RFID reader, MongoDB database, NativeScript application and mobile devices). Therefore, two networks had to be created. The first network (LAN) connects the development laptop (MacOS), where both the NativeScript application and MongoDB database are running, to the Windows PC, where Matlab is running. Due to the use of .NET libraries in Impinj Octane SDK, Matlab had to be installed on a Windows PC. Another solution would be to run Windows on a VM on the MacOS developer machine (MacBook Pro Spring 2015, operating system: macOS High Sierra 10.13.5, RAM: 8 GB, Internal Storage: 128 GB), but unfortunately the RAM as well as disk space restricted the functionality of the VM. The second network that was installed was a private WLAN which connected all mobile devices (smartphones and tablets) with the MacOS developer machine. This connection enables receiving and transfering all data from the MongoDB database into the mobile app. To be precisely, the WLAN provides a secure connection to all mobile devices by using a Wi-Fi Protected Access 2 (WPA2).

3.3.1.1.- Challenges during development

During the development, many difficulties and challenges had been faced. The following section focusses on some problems during development. To start with, the setup of the development environment of a NativeScript application was complicating because for example on Windows computers there cannot be installed XCode. So, the only solution to develop and run the application both on Android and iOS devices was to use a MacOS computer or to execute the application on Nativescript Sidekick (using its cloud build service).

Another challenge was the real-time synchronization of the mobile application with MongoDB server was difficult. Firstly, a HTTP request/response was implemented to connect the mobile device with the RFID reader and the database. Furthermore, refreshing the page was not automated. Thus, another solution had to be considered. Finally, a Socket.IO connection was implemented between both server and client. Socket.IO enables automatic two-way data synchronization and cannot be compared to HTTP Request/Response. By opening a Socket connection between the server and all clients, it was possible to run the application on various mobile devices without any complication.

After that, another complication was the connection between MongoDB, Matlab and the NativeScript application. This was solved by using a server instance which both connects the database as well as the Nativescript application. Section 3.3.2 explains the exact implementation and the used architecture to connect all system components.

3.3.1.2.- User Scenario

Figure 3.5 gives an example of a user scenario. The figure shows two treatment rooms in a hospital, both equipped with a RFID antenna. In the hallway, there are two additional antennas which are connected to the RFID reader. When passing one room, the nurse or doctor will roll a trolley with the needed drugs (all marked with a RFID tag). If the trolley is near to the RFID antenna #4, it will send the current drug information to the RFID reader which synchronizes them with the computer and the tablet or smartphone. When entering a treatment room, RFID antenna #3 or #2 receive the tag's information and synchronize them similarly as before.

At the bottom of the figure, there can be seen a intranet/internet connection. This means, that the system can be both set up in the local intranet (which enables a higher privacy and security) as well as in the internet. In the end, each RFID antenna can signify a different action, e.g. antenna #4 = 'medication will be administered', antenna #3 and #2 = 'medication is dosed to patient X', #1 = 'medication has been dosed successfully'.

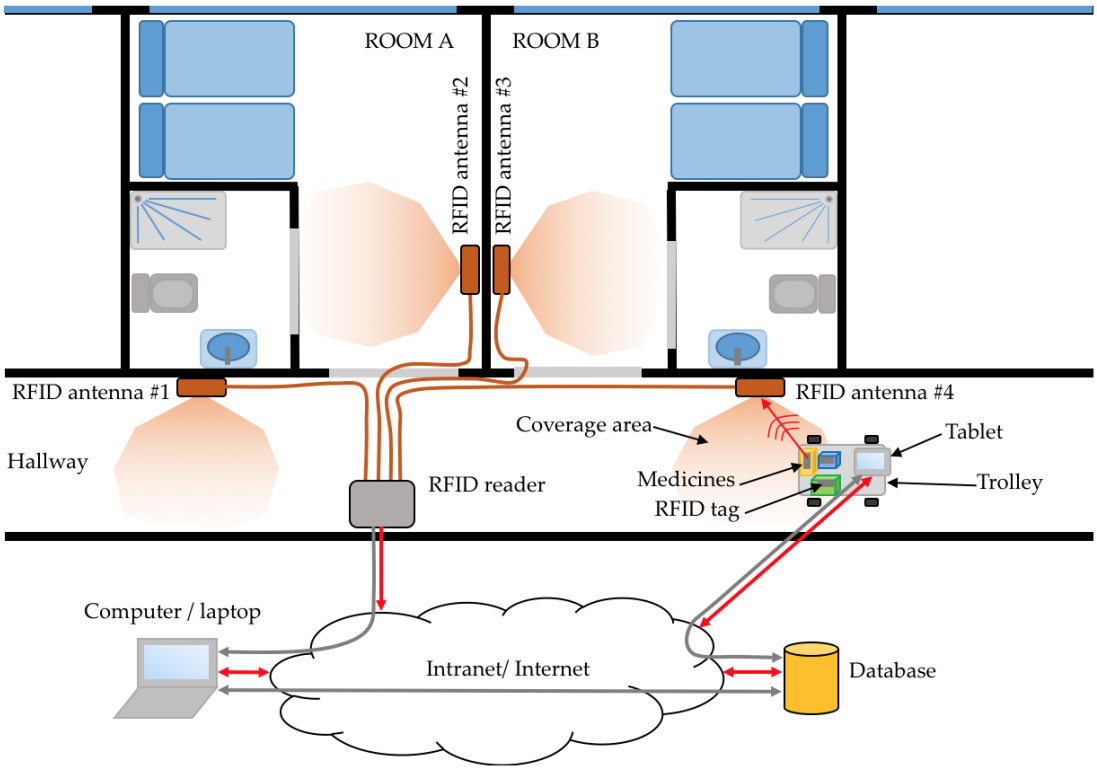
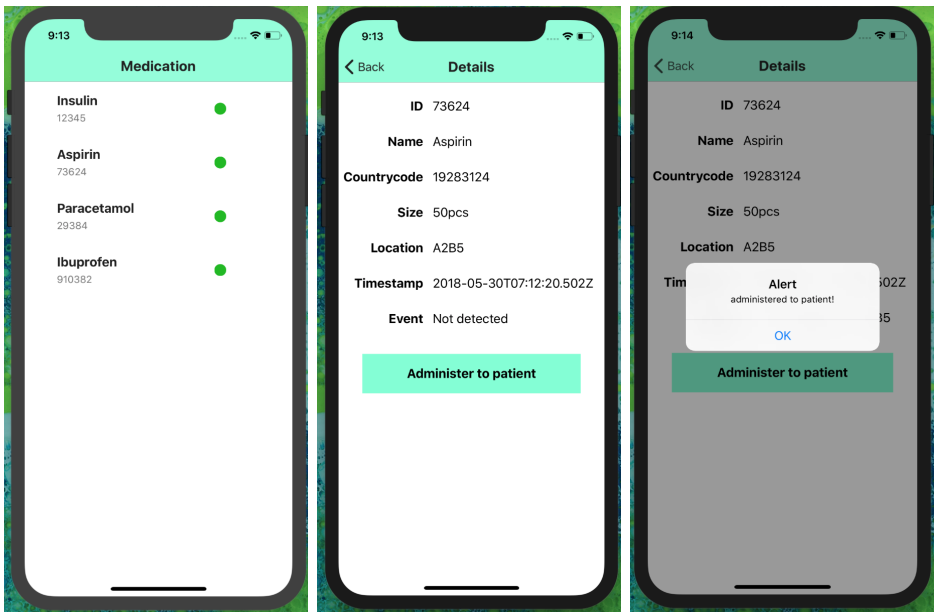


Figure 3.5.- User scenario of RFID application

Figure 3.6 shows three screenshots of the developed application. The left picture shows the start screen containing a list of all drugs which were detected by the RFID reader and are stored in the database. The screenshot in the middle shows the screen after selecting one drug of the start page. It gives further information of the selected item. Each drug is stored with an 'ID' (which is the same as the RFID tag number). The 'Countrycode' attribute refers to the country-specific number of each drug. For example, in Germany 'Aspirin' has a specific unique code which is called Pharmazentralnummer (PZN), and contains a 7- or 8-place number (e.g. 04114918). The 'Location' attribute describes the exact location of a drug and refers to the location system in the stock (e.g. in pharmacies). The information 'Timestamp' refers to the last time a drug was detected. The value on the right side of 'Timestamp' is the default value for storing timestamps in MongoDB. If another date format is required, this can be easily defined in the MongoDB schema definition. The last information about the selected drug, named 'Event', represents the information about a drug being detected by one of the antennas or administered to a patient. If the nurse or doctor selects the



(a) Start page of application

(b) Item information page

(c) Information about administered drug to patient

Figure 3.6.- Layout of application, Screenshot of iOS Simulator

button 'Administer to patient', the system checks, if the selected drug is permitted to be administered to the patient in the room or not. If it is permitted, the information about the administered drug will be stored in the database and the content of the 'Event' changes to 'Administered to patient in room XX'.

3.3.2.- Software Architecture

The following section focusses on the system's architecture which can be seen on figure 3.7. In the top right corner, the drugs and patients which can be tagged with a RFID tag are displayed. When approximating an antenna of a RFID reader, information about medication (e.g. stock availability ect.) are transmitted. At the same time, the RFID reader registers the detected tag and captures its information using Matlab. Matlab synchronizes the transmitted information with the MongoDB database. When receiving current drug information, the server instance sends the data through a Socket.IO connection to the client (smartphone or tablet). Synchronously, all data is transmitted and displayed on the mobile device.

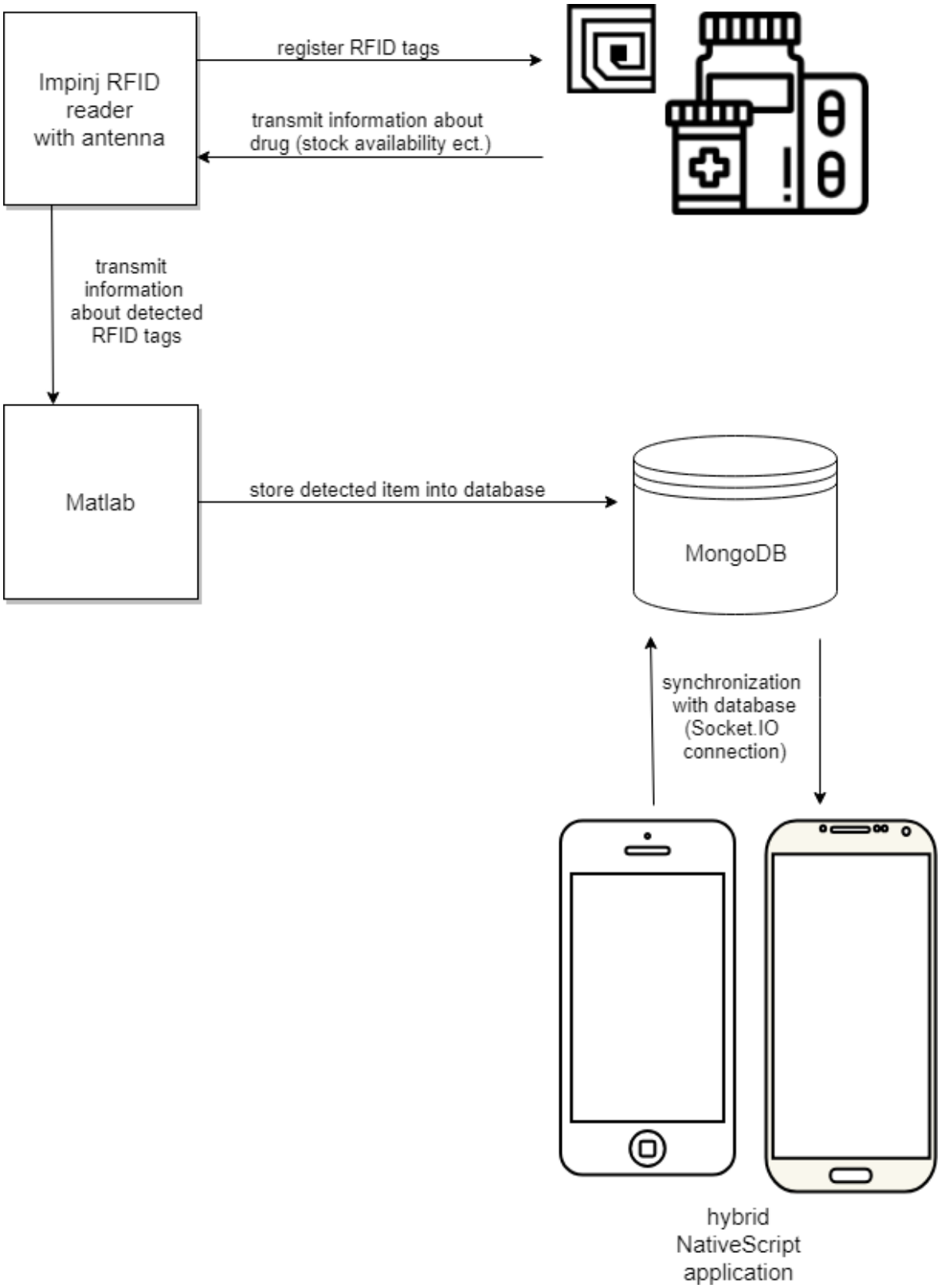


Figure 3.7.- The developed system architecture of the mobile RFID application

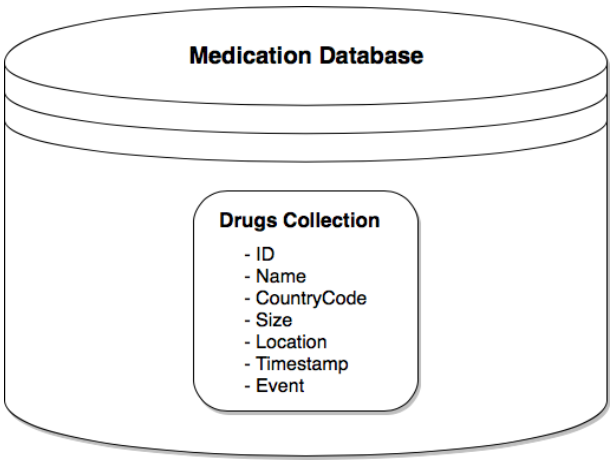


Figure 3.8.- Applied data model

3.3.2.1.- Datamodel

Figure 3.8 reflects the outline of the developed MongoDB document related to its context and the MongoDB database. As depicted in Figure 4.6, each drug entry contains information about its ID, name, countrycode, size, location, timestamp and event. All documents are pooled in a drug collection which forms a part of medication database. This data model can be extended for example by other collections containing information about several equipment or patients.

3.3.2.2.- LAN Architecture

Figure 3.9 shows the scheme of the connections between all relevant components into one LAN. There exist two networks: On the one hand, the LAN between RFID reader, laptop and desktop PC. On the other hand, a local network between the laptop and various mobile devices, like smartphones or tablets is established. The only connecting point between both networks is the laptop with the IP 169.254.1.2.

At the top of figure 3.9, there is a '5-Port-Ethernet-Hub' which enables connecting up to five different devices or computers via ethernet. On the left of the picture, the RFID reader, with IP 169.254.1.1, is depicted. Each time, when detecting a RFID tag, the RFID reader sends the information to Matlab which is running on the desktop PC with the IP 169.254.1.3.

After receiving the detected RFID tag's information, the desktop PC transmits the data to MongoDB database running on the laptop with the IP 169.254.1.2. The purpose of the second local network between laptop and mobile devices is to execute the NativeScript app and to visualize currently detected RFID tags in the app.

3.3.2.3.- Reading Process in Matlab

The following section focusses on the reading process which is implemented in Matlab. Matlab uses the Impinj Octane SDK [18] which easily connects the RFID reader to Matlab. The usual reading process in Matlab can be seen in figure 3.10. In the top left corner of figure 3.10, after starting the Matlab code, a setup method triggers the initialize method.

After initializing, Matlab starts reading tags during a predefined time (t). Every time a RFID tag is read, the method eventhandlerChanged() is executed. EventhandlerChanged() notices the changes in the detected tags and triggers a new action 'store data'. After that, analogous to the first concept of the system (see section 3.1.3), the *.mat data is converted into *.csv data and as such exported. In the most current version of the system, the two last actions (conversion and export) are brought together and data is directly transfered into the MongoDB database.

3.4.- Tests of system

Hospital Universitario Central de Asturias (HUCA) which is situated in Oviedo, was created in 13 December 1989 and depends on the 'Servicio de Salud del Principado de Asturias' [19]. Since 28 February 1990, HUCA has the character of an University Hospital due to the cooperation with the university of Oviedo as well as 'Insalud' and Principado de Asturias, established in 'la Orden'. HUCA provides 944 beds, 25 operating suites, 238 rooms for external surgical as well as two gamma cameras (Single-photon emission computed tomography (SPECT)/Technetium (TC)) [19]. Altogether, HUCA is divided into 43 medical subjects, such as alergology, clinical analysis, clinical

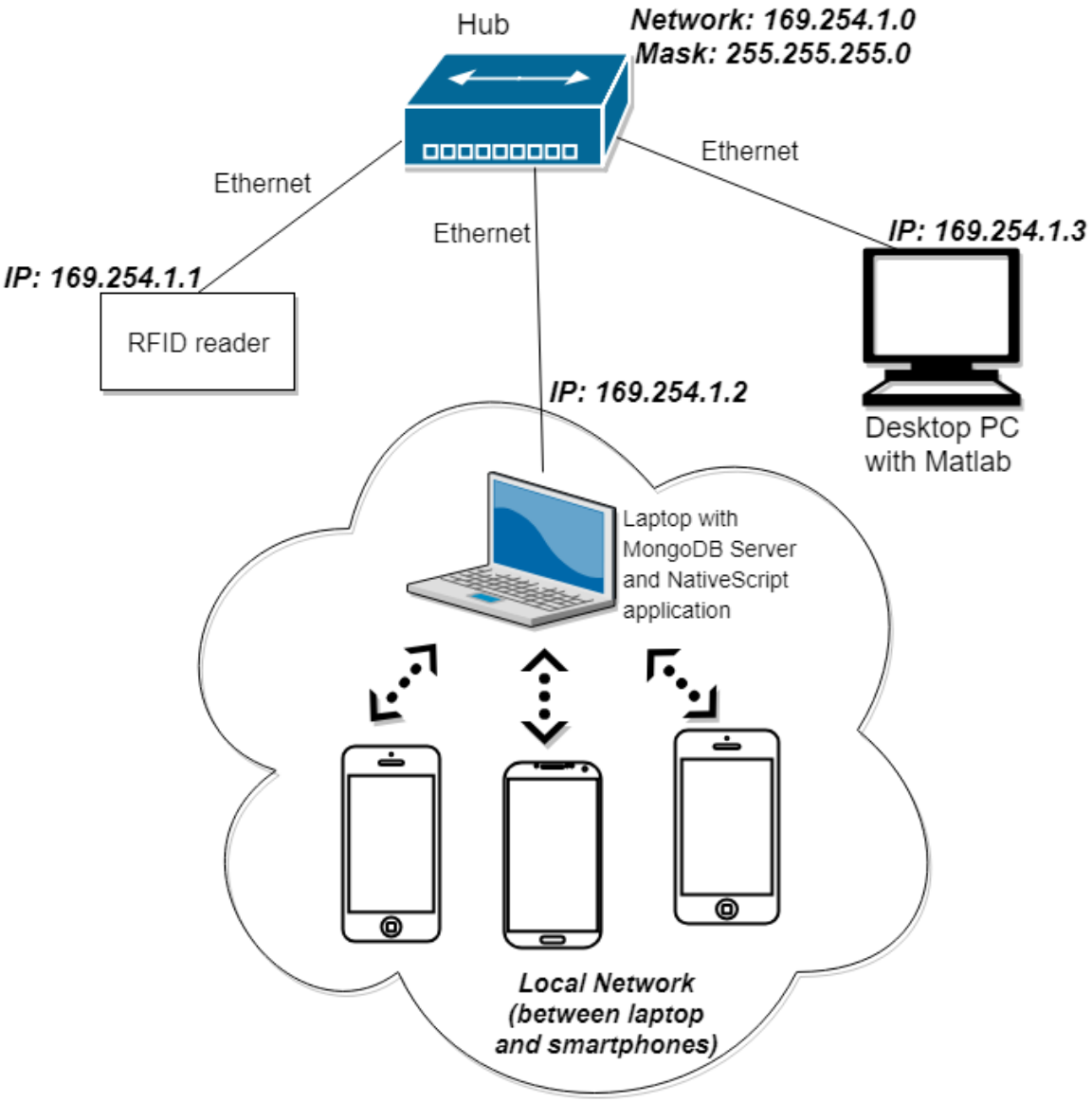


Figure 3.9.- The LAN architecture of the developed system

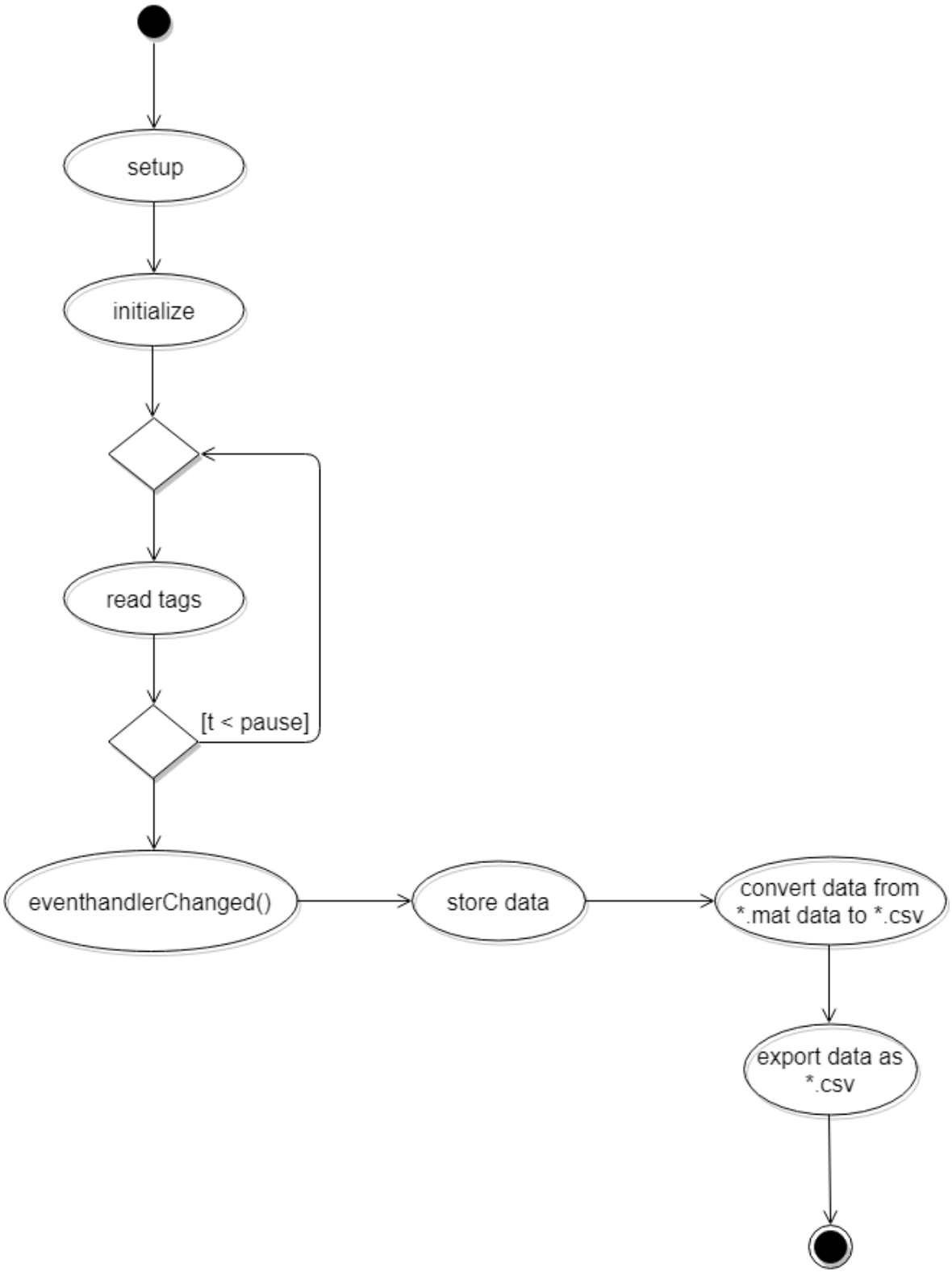


Figure 3.10.- Activity diagram: reading process in Matlab

biochemistry, surgery, Nephrology, Neurology, Rehabilitation, prevention of work-related risks, radiodiagnostics etc.

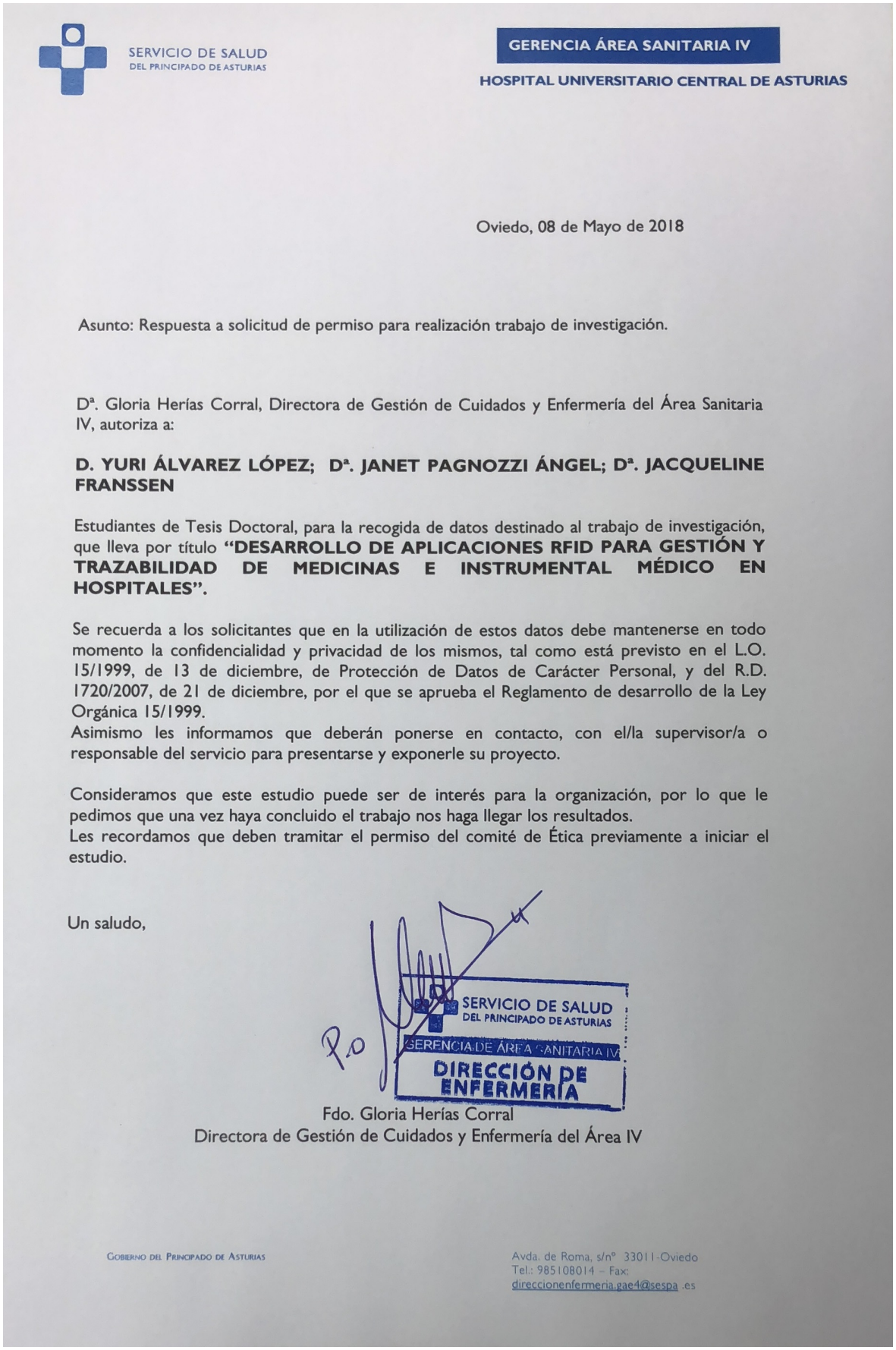
Besides, HUCA forms a part of the 'unit of national reference for work-related illnesses of the respiratory system'. Together with the 'Hospital Monte Naranco', 'hospital del Área Sanitaria IV' and with reference to Servicio de Salud del Principado de Asturias (SESPA) 'Servicio de Salud del Principado de Asturias' and finally the 'Instituto Nacional de Silicosis', HUCA promotes research on the subject of respiratory disease and its preventions measures.

Figure 3.11 shows the permission letter, signed by the director Gloria Herías Correal of HUCA on 8 May 2018 in Oviedo. The tests are planned for June 2018, an exact date is not fixed already (effective 18 May 2018).

3.4.1.- Planned tests of system

3.4.1.1.- Test scenario

The following section describes a test scenario which is planned to be administered in HUCA. To begin with, given the situation of a nurse which has to administer several drugs to many patients in a hospital. In the past, there where many cases, where nurses dismissed or administered medication to patients incorrectly. To prevent those errors and to improve patient's safety, the nurse can control and capture each time she is administering a drug to a patient. The system will check if the selected drug (which was detected in the patient's room) is valid and if it can be administered to the patient. For example, the nurse is walking with a transport trolley (containing several drugs) on the hallway of a department in the hospital. On the hallway, there are two antennas which are connected to a reader installed nearby. When passing these antennas, the detected drugs will be automatically stored into the MongoDB database and displayed on the mobile phone or tablet. When entering the patient's room, a second antenna detects the drugs and transmits the signal to the reader which receives the information about the location of the drug ('drug X detected in room 314'). After that, if the nurse wants to administer a drug, the system checks whether it was already administered



SERVICIO DE SALUD
DEL PRINCIPADO DE ASTURIAS

GERENCIA ÁREA SANITARIA IV
HOSPITAL UNIVERSITARIO CENTRAL DE ASTURIAS

Oviedo, 08 de Mayo de 2018

Asunto: Respuesta a solicitud de permiso para realización trabajo de investigación.

Dª. Gloria Herías Corral, Directora de Gestión de Cuidados y Enfermería del Área Sanitaria IV, autoriza a:

D. YURI ÁLVAREZ LÓPEZ; Dª. JANET PAGNOZZI ÁNGEL; Dª. JACQUELINE FRANSSEN

Estudiantes de Tesis Doctoral, para la recogida de datos destinado al trabajo de investigación, que lleva por título **"DESARROLLO DE APLICACIONES RFID PARA GESTIÓN Y TRAZABILIDAD DE MEDICINAS E INSTRUMENTAL MÉDICO EN HOSPITALES".**

Se recuerda a los solicitantes que en la utilización de estos datos debe mantenerse en todo momento la confidencialidad y privacidad de los mismos, tal como está previsto en el L.O. 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal, y del R.D. 1720/2007, de 21 de diciembre, por el que se aprueba el Reglamento de desarrollo de la Ley Orgánica 15/1999.
Asimismo les informamos que deberán ponerse en contacto, con el/la supervisor/a o responsable del servicio para presentarse y exponerle su proyecto.

Consideramos que este estudio puede ser de interés para la organización, por lo que le pedimos que una vez haya concluido el trabajo nos haga llegar los resultados.
Les recordamos que deben tramitar el permiso del comité de Ética previamente a iniciar el estudio.

Un saludo,

SERVICIO DE SALUD
DEL PRINCIPADO DE ASTURIAS
GERENCIA DE ÁREA SANITARIA IV
DIRECCIÓN DE ENFERMERÍA

Fdo. Gloria Herías Corral
Directora de Gestión de Cuidados y Enfermería del Área IV

GOBIERNO DEL PRINCIPADO DE ASTURIAS

Avda. de Roma, s/nº 33011-Oviedo
Tel.: 985108014 - Fax:
direccionenfermeria.gae4@sespa.es

Figure 3.11.- Permission letter about application tests in HUCA

and whether it is permissible. After having administered the selected drug, the event will change to 'already administered in room X' and will be stored in the database.

3.4.1.2.- Functional tests

In order to test the real-time synchonization, all entries in the database were deleted and the application was refreshed (by using the pull-to-refresh plugin). After five seconds, the alert dialog which explains that there are no entries in the database and no detected drugs is displayed (tested 06 June 2018).

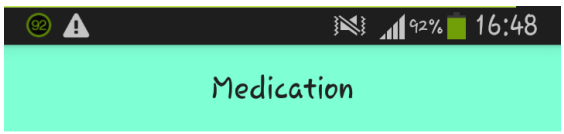
3.4.1.3.- Device test

The application was tested on four devices including both tablets and smartphones. During development, mainly the iOS simulator was used to rapidly execute and synchronize the application. Since installing a prototype app on an iOS device requires several developer certificates and some configuration steps, it was easier to use the included simulator which is offered by XCode. Later, when the app could be executed without throwing any errors, it was installed and tested on an iPhone 7 (Model: 15F79, iOS 11.4). Furthermore, it was installed on an iPad (iOS 11.4) and ran successfully. Moreover, the app was tested on two Android devices: a smartphone (Samsung Galaxy S3, Model: GT-9300, Android 4.3) as well as a tablet (Model: Samsung GT-P5110, Android 4.2.2, screen size: 10.1"). The successful test results can be seen in figure 3.12. The figure shows the screenshots of the smartphone "Samsung S3 GT I9300".

3.4.2.- Test results

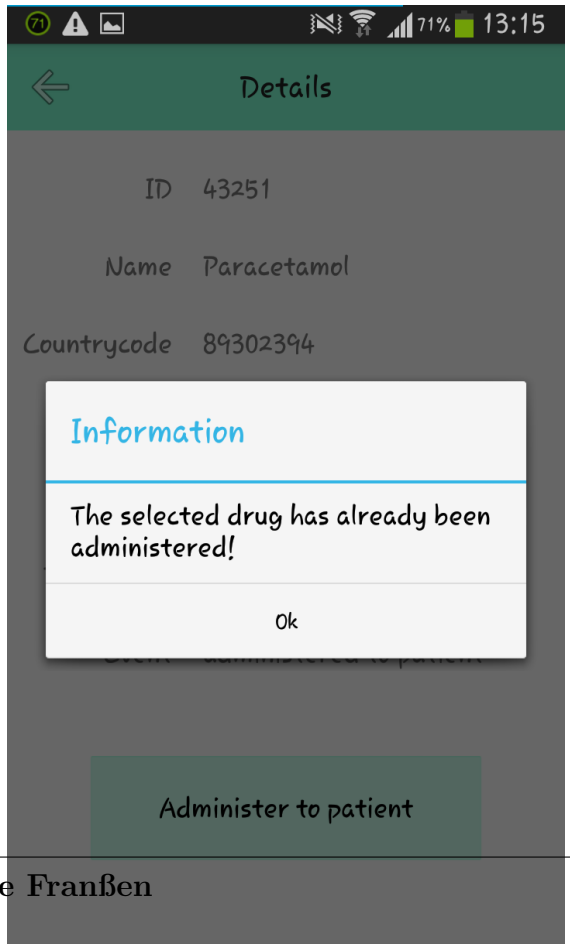
3.4.3.- Evaluation of test results

During the first tests in the HUCA on Monday, the 25 of June 2018, some small features of the application were noticed negatively. For instance, the "real-time" synchronization between the mobile device and the database. The reading process of the RFID reader and the following insertion process into the database did work without problems, but the Socket.IO connection between the mobile client and the


92%
16:48
Medication


72%
13:14
Medication
Medicina Buena
92783
Paracetamol
43251
Tesafilm
98374
Chupachups
82749
Chocolate negro
34215
Chocolate blanco
73829
Chocolate leche
82913


71%
13:15
Details
ID 43251
Name Paracetamol
Countrycode 89302394
Size 50pcs
Location B34C2
Timestamp 2018-06-07T11:15:02.7532
Event administered to patient
Administer to patient


71%
13:15
Details
ID 43251
Name Paracetamol
Countrycode 89302394
Information
The selected drug has already been administered!
Ok
Administer to patient

database server has to be optimized. Moreover, the feature of pulling down the app to refresh all data turned out to be rather obstructive than useful. To solve this problem, a function which automatically reloads the app's main page has to be implemented. These two features will be optimized until the next test in the HUCA until the 04 or 05 of July.

3.5.- Summary and Outlook of application

The developed application can be extended in many ways. In the following section the three most important extension points will be outlined.

During testing in the HUCA, a nurse mentioned that when administering drugs to patients, she usually has to carry a medicine trolley with several containers, each for a specific patient. Actually, most of the patients in a hospital have to take more than only one drug which has to be organized in a list. In the past, the nurses had to manually check the patient related medication on a paper. So, in future it would be very useful to divide the application into different sections which are related to a specific patient. This also implies, to change the format of the stored data. To be precisely, each patient could be stored in the MongoDB database as a top level document and contains various subdocuments which are his drugs. Since the structure of these drug documents already exists in the developed application, it would be very easy to adopt and implement that solution. Another possibility would be to connect the application to each electronic patient record, which can be seen in figure 3.13. Consequently, each patient can be easily identified with its medication list.

Furthermore, in case of a patient's room where two or more patients are accomodated, the system should distinguish between those. To fastly identify a patient and its related medication on the medicine trolley, one solution could be to let patients wear a wristband with the same passive RFID tag as the drugs. Each time, the same antenna detects a drug and the patients wristband, it checks whether the drug should be administered to the patient and either allows the administration or not.

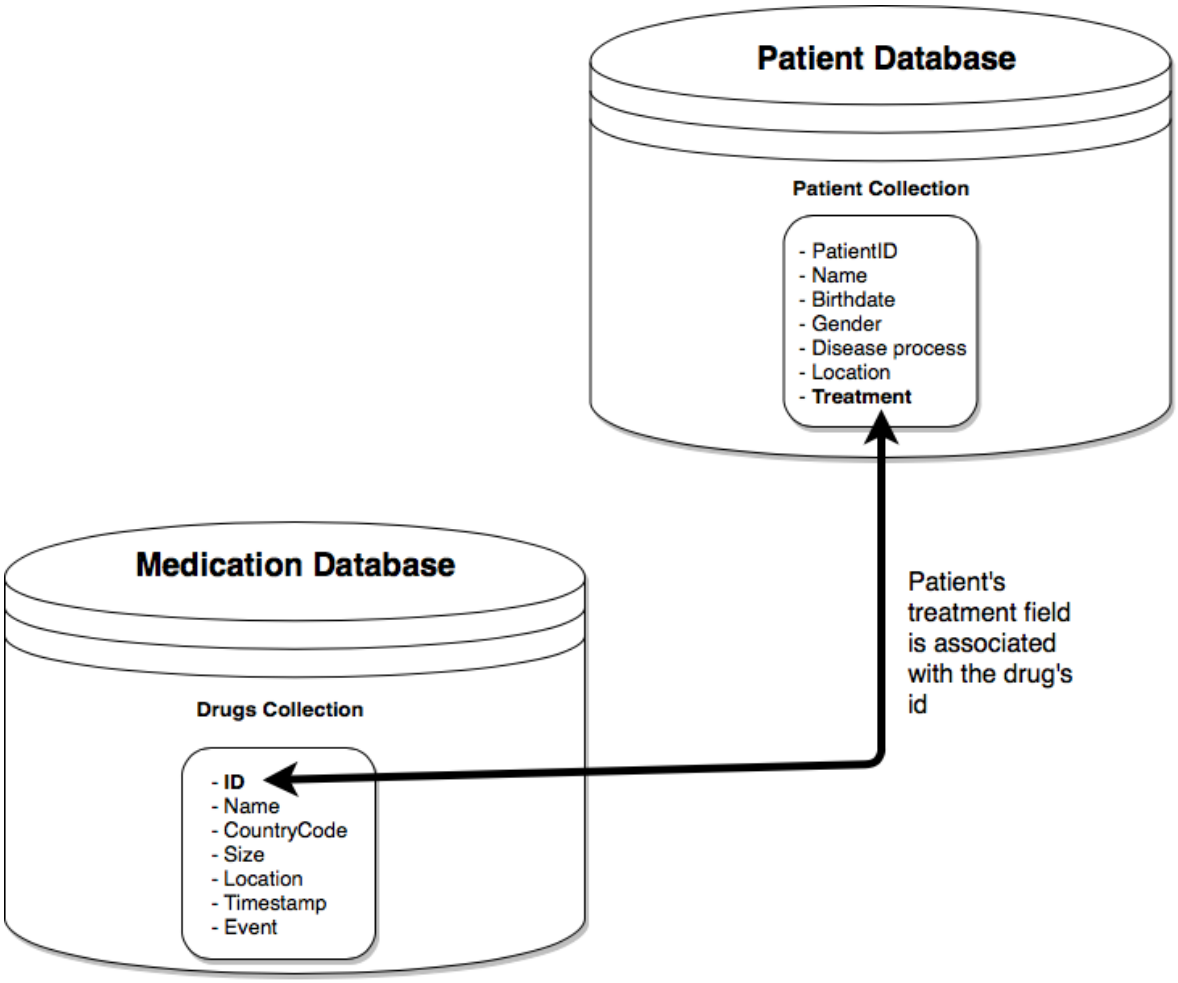


Figure 3.13.- Example of extended data model, each patient entry is associated to the medication list

Another possible scenario would be to set up user roles in the application. This means, that for example the chief physician or the head of nurses sets up a list of medication related to each patient and only this person has access to the personal data. To provide patient's safety, there should be no other nurse or stuff in the hospital who can access the patient's related medication data. After setting up the drugs and its permissions to be administered to the specific patients, this "administrator" saves all data and logs out. Apart from this administrator, there will be another account, such as "nurse" or any ordinary user account. This person administers the drugs to each patients.

Bibliography

[1] D. Henrici, *RFID security and privacy: concepts, protocols, and architectures*, ser. Lecture notes electrical engineering 17. Berlin: Springer, 2008, 269 pp., OCLC: 244058698.

[2] G. Tamm and C. Tribowski, *RFID*, ser. Informatik im Fokus. Berlin: Springer, 2010, 144 pp., OCLC: 845690868.

[3] S. Ajami and A. Rajabzadeh, "Radio frequency identification (RFID) technology and patient safety", *J Res Med Sci*, vol. 18, no. 9, pp. 809–813, Sep. 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3872592/ (visited on 03/12/2018).

[4] S.-W. Wang, W.-H. Chen, C.-S. Ong, L. Liu, and Y.-W. Chuang, *RFID Application in Hospitals: A Case Study on a Demonstration RFID Project in a Taiwan Hospital*. 2006, vol. 8, 184a–184a. DOI: 10.1109/HICSS.2006.422.

[5] *Mobile Computing und RFID im Facility Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-77552-2. [Online]. Available: http://link.springer.com/10.1007/978-3-540-77552-2 (visited on 03/13/2018).

[6] R. Rezaiesarlak and M. Manteghi, *Chipless RFID*. Cham: Springer International Publishing, 2015. DOI: 10.1007/978-3-319-10169-9. [Online]. Available: http://link.springer.com/10.1007/978-3-319-10169-9 (visited on 03/13/2018).

[7] (Feb. 19, 2010). Medical uses for RFID products, Vizinex RFID, [Online]. Available: https://www.vizinexrfid.com/medical-uses-for-rfid-products/ (visited on 03/07/2018).

[8] A. Meier and M. Kaufmann, *SQL- & NoSQL-Datenbanken*, ser. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. DOI: 10.1007/978-3-662-47664-2. [Online]. Available: http://link.springer.com/10.1007/978-3-662-47664-2 (visited on 04/26/2018).

[9] S. G. Edward and N. Sabharwal, *Practical MongoDB*. Berkeley, CA: Apress, 2015. DOI: 10.1007/978-1-4842-0647-8. [Online]. Available: http://link.springer.com/10.1007/978-1-4842-0647-8 (visited on 04/26/2018).

[10] Á. Rocha, H. Adeli, L. P. Reis, and S. Costanzo, Eds., *Trends and Advances in Information Systems and Technologies: Volume 2*, Advances in Intelligent Systems and Computing, Springer International Publishing, 2018. [Online]. Available: / / www . springer . com / de / book / 9783319777115 (visited on 05/04/2018).

[11] (May 16, 2018). MongoDB connection - MATLAB - MathWorks españa, [Online]. Available: https://es.mathworks.com/help/database/ug/mongo.html (visited on 05/16/2018).

[12] (May 10, 2018). NativeScript, NativeScript.org, [Online]. Available: https://www.nativescript.org/ (visited on 05/04/2018).

[13] *Nativescript-cli: Command-line interface for building NativeScript apps*, original-date: 2014-06-30T10:21:20Z, Mar. 7, 2018. [Online]. Available: https://github.com/NativeScript/nativescript-cli (visited on 03/08/2018).

[14] (May 4, 2018). Play and try NativeScript on your device – {n} playground, [Online]. Available: https://play.nativescript.org/ (visited on 05/10/2018).

[15] (May 10, 2018). NativeScript marketplace, [Online]. Available: https://market.nativescript.org/ (visited on 05/10/2018).

[16] (May 11, 2018). Speedway r420 RAIN RFID reader — impinj, [Online]. Available: https://www.impinj.com/platform/connectivity/speedway-r420/ (visited on 05/11/2018).

[17] (May 17, 2018). An introduction to MATLAB, IEEE UFFC, [Online]. Available: https://ieee-uffc.org/ultrasonics/software/an-introduction-to-matlab/ (visited on 05/17/2018).

[18] (May 16, 2018). Octane SDK, Impinj Support Portal, [Online]. Available: http://support.impinj.com/hc/en-us/articles/202755268-Octane-SDK (visited on 05/16/2018).

[19] (May 18, 2018). HUCA, [Online]. Available: http://www.hca.es/huca/web/index.asp (visited on 05/18/2018).