Universidade de Aveiro
2015
Departamento de
Eletrónica, Telecomunicações e Informática

Tiago Miguel Vila Flor Lourenço

Uma aplicação Web para apoio do diagnóstico e monitorização automática de Septicemia

A web application to support an automated diagnostic and monitoring of Sepsis

Universidade de Aveiro
2014
Departamento de
Eletrónica, Telecomunicações e Informática

Tiago Miguel Vila Flor Lourenço

Uma aplicação Web para apoio do diagnóstico e monitorização automática de Septicemia

A web application to support an automated diagnostic and monitoring of Sepsis

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor José Luís Oliveira, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor David Campos da empresa BMD Software.

Aos meus pais.

o júri / the jury

presidente / president

Prof. Doutor Joaquim Manuel Henriques de Sousa Pinto
Professor Auxiliar do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor António Manuel de Jesus Pereira
Professor Coordenador do Departamento de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

Prof. Doutor José Luís Oliveira
Professor Associado do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro (orientador)

agradecimentos / acknowledgements

Desde já, gostaria de agradecer aos meus orientadores, José Luís Oliveira e David Campos, por todo o apoio e conselhos ao longo desta etapa.

Não posso deixar de agradecer a todos aqueles que me ajudaram durante o meu percurso académico, com especial apreço à Ana Domingos e ao Tiago Soares.

Aos meus pais Carlos Lourenço e Zita Lourenço, por me terem apoiado ao longo de todos estes anos. A eles dedico esta dissertação.

A todos, saibam que vos agradeço por tudo.

palavras-chave

Plataforma web, informação em tempo real, interface user-friendly, controlo de dispositivos

resumo

Spesis é uma doença que resulta numa infeção geral do organismo, causada por agentes patogénicos. É uma enfermidade com uma elevada taxa de mortalidade, a maior parte das vezes resultante de diagnóstico tardio. Isto ocorre porque a sua deteção muito é, em geral, muito demorada em comparação com a velocidade de evolução da doença. Por outro lado, o processo de deteção implica que sejam feitas várias análises ao sangue do paciente, as quais são feitas por departamentos distintos e a diferentes taxas de execução. Até todas as análises serem concluídas para posteriormente serem visualizadas pelos médicos, a doença continua a progredir. A solução para este problema, passa por métodos de diagnóstico mais eficazes e atempados, bem como a troca de informação sobre o estado do paciente em tempo real, ao longo de todo o processo.

Nesta dissertação vamos apresentar uma plataforma web que é responsável por fornecer toda a informação atualizada aos vários utilizadores envolvidos, os quais podem obter em tempo real a informação associada à análise de um paciente e ao seu estado atual. O fornecimento contínuo de informação garante que os vários utilizadores tomem decisões mais informadas em relação ao tratamento do paciente, permitindo uma taxa de eficiência superior à atual.

keywords Web platform, real-time information, user-friendly interface, control devices

abstract

Spesis is a disease that generally results from an infection of the organism, caused by pathogens. It is a disease with a high mortality rate, most often resulting from a late diagnosis. This happens because his detection is much slower in contrast to it’s rate of evolution. On the other hand, the detection process requires several analyzes to be made from the patient's blood, which are done by different departments at different implementation rates. Until all analyzes are completed, for a later analysis by the doctors, the disease continues to progress. The solution to this problem involves methods more effective and timely diagnosis, and also the exchange of information about the patient's condition in real time, throughout all the process.

In this thesis we present a web platform that is responsible for providing all the updated information to the various users involved in the process, which can get real-time information associated with the analysis of a patient and their current state. The continuous supply of information ensures that all the users can make more informed decisions regarding the best treatment for patient, allowing a higher efficiency rate than the current one.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis presents the main ideas and basic stages of design and development of a web platform called HemoSpec Platform. This software will be developed by BMD Software, an SME located in Aveiro, under the European Union project HemoSpec.

HemoSpec will feature a medical web platform containing all the relevant information related with patients and allowing users to take right decision, at the right time. This platform's main aim is allowing a fast and reliable diagnosis of the sepsis disease, leading to an early and more effective treatment, consequently improving the current rate of success. It will include a multiplex photonic surrogate process that simultaneously analyses biomarker in the plasma and probes the biophotonic fingerprint of blood cells [1].

HemoSpec's web platform will be the front end for each player involved in the patient's treatment workflow. This platform will allow managing patients, exams and diagnostics in a unique console. For each diagnostic requested, the application backoffice will trigger exams and laboratory studies. The resulting data is then integrated and submitted by the developed classification models that will lead to the final diagnostic. Each player will have instant access to the patient's data and workflow. This improves the user interaction with the patient's process, granting the ability to take all necessary actions through a browser (Chrome, Mozilla Firefox, etc.), that allows managing the patient's treatment workflow. Users involved in the patient's workflow are notified when the patient information changes. This

communication mechanism ensures a more reliable and immediate information exchange process between different users.

To correctly understand all system's workflow and all kinds of functionalities that the platform supports, as well as if it fulfills all requirements, a detailed list of the proposed requirements, the mockups obtained from requirements and the final user interface will be presented.

1.1. Motivation

Currently, there is still a lack of information about Sepsis disease around the world [1]. Therefore, the treatment priority is low while the disease evolution is increasing dramatically, as well as the growing mortality rate of the disease. In addition, a clear diagnosis can be lengthy and complicated, due to highly variable symptoms and non-specific criteria, causing the disease to be diagnosed and treated too late (Figure 1).

Beyond the mentioned restrictions, other big hindrances are the procedures needed to confirm the diagnosis. Initially, it is necessary to collect four patient's blood samples which will be distributed within the different laboratories to perform specific analysis. After this, each laboratory writes a report (on paper format) and sends it to the technician responsible for gathering and forwarding them to the physician, who will then analysis and take a decision according to the patient's condition. However, during this process failures can occur, since liability and data security cannot be guaranteed during the analysis process. A critical situation is the extensive waiting time. Physicians have to wait by all reports before making a decision, which can affect its accuracy considering that some reports might already be outdated.

This time consuming process affects the disease's development, causing the treatment to be applied too late, reducing the rate of success.

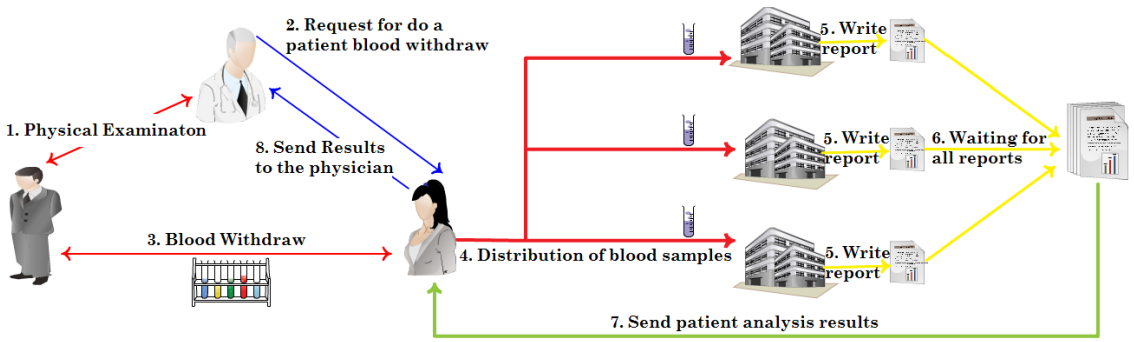

Figure 1 – Actual sepsis treatment

Patients are the most harmed by this long workflow, since they depend exclusively on the diagnostic speed. Even though they don't interact directly with the system, they are considered to be its central element.

Currently, "the sepsis mortality range is from 7% within the less severe cases to almost 50% in case of septic shock" [1]. This value clearly implies the necessity of an early diagnosis and severe treatment. In that sense, the main goal of the HemoSpec system is to drastically reduce the mortality range, enhancing the treatment's success.

Summarizing, we can conclude that the current sepsis diagnosis process is slow and complex, implying several steps from different actors, which affects the patient's treatment. While occurs the process of collecting the blood, writing the reports and taking them to the respective physician, many complications may arise such as loosing or mixing documents that retarding the diagnosis. The current used methodology has not proven to be efficient and reliable, seriously causing damages to the involved patients and contributing to the increase of the sepsis' mortality rate.

To address this problem the HemoSpec system is being developed, consisting of two main components: HemoSpec Platform and HemoSpec Device.

The HemoSpec Platform allows the physician to create and start a patient's workflow, where we can see all information about on-going processes. After starting the workflow, the platform notifies technicians that a new case is ready to be analyzed. Then, a technician places just one patient's blood sample (1-2ml) in the HemoSpec device, avoiding the distribution of

several samples like in the actual process, which is also less harmful to the patient. To achieve this, HemoSpec Device involves "the development of three modules for rapid detection of four biomarkers to advance the diagnosis and management of sepsis: automated micro-fluidic sample handling, simultaneous multiplex fluorescence biomarker sensing and detailed Raman spectroscopic leukocyte characterization" [1]. After, the device alerts if the sample was well inserted and the technician can start the analysis process, having full control of it, which means he can interact instantaneously by starting, stopping or resetting it.

When this workflow part is finished, the device will then present the results, which will be immediately available to all the actors involved in the patient's treatment. When the technician validates the results, the HemoSpec Platform stores them and performs a background data-mining process that allows concluding the disease classification based on real facts. According to this classification, new information is added to the patient's profile. HemoSpec centralizes all information about patient's disease treatment and shows this data on HemoSpec Platform. Finally, physicians can visualize the final results, in real time, ensuring an anticipated patient's diagnosis and suitable treatment (Figure 2).

Therefore, our solution intends to reduce the procedure time, the risk of errors within the process of exchanging data and guarantee that the patient receives on time, the proper treatment according to his needs. This will also lead to a reduction of the mortality rate and related treatment costs.

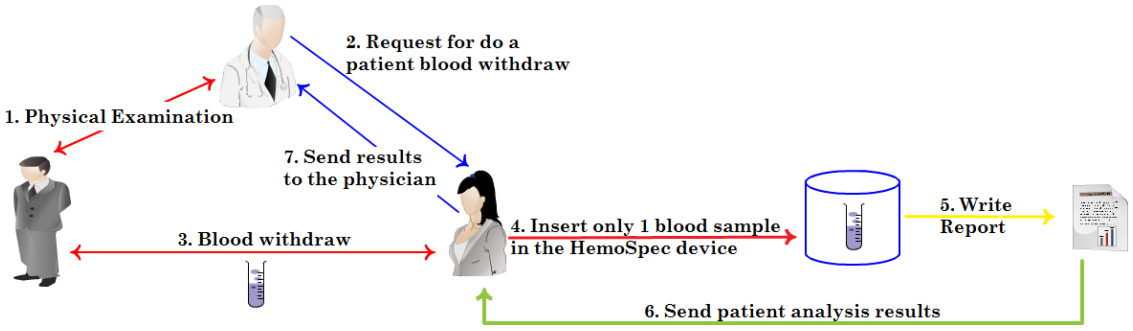

Figure 2 – Integrate HemoSpec project in sepsis treatment

1.2. Goals

The aim of this thesis focuses mainly on the following points: the first goal is the development of a web platform to incorporate in the HemoSpec project. As a second objective it has the intention to create a set of web services that permits external clinical and diagnostic modules to communicate automatically with this system. The last objective is to develop a communication protocol that allows to synchronize data and actions to be performed between HemoSpec Platform and the HemoSpec Device, ensuring updated data.

From a top down approach and to achieve the goals, it will be necessary to go through the following steps:

- Project requirements analysis;
- Mockups;
- System architecture design;
- Develop a communication protocol;
- Understand user interface design patterns;
- Client/Server development (HTML5, CSS3, Javascript, Web Services …).

1.3. Thesis outline

The remaining chapters of this thesis are organized according to the following:

Chapter 2 lists a set of system's requirements, like user requirements that guided the development of our solution, and detects limitations and issues that needs to be covered. We present a list of use cases and respective diagrams that shows necessary actions supported by the platform. Also, we present an activity diagram that describes a basic workflow of the solution. Finally, we show the mockups obtained after analyzing all requirements.

Chapter 3 presents HemoSpec architecture. In this section we start by presenting the overall architecture of the solution. After, we describe specific information related with the client-side and server-side of the application. In the server side section, we describe the communication protocol that ensures

all data exchanges, and the actions that occurs inside of the HemoSpec project.

Chapter 4 presents the overall implementation of the designed solution. Firstly, we present the system description. Secondly, we describe the implementation of the communication protocol and finally we show and explain the HemoSpec Platform's user interface.

Chapter 5 presents an analysis of the user testing plan to validate the usability and the proper functioning. Results are also shown.

Chapter 6 presents some conclusions about the work done, as well as future work that can be done in order to improve the developed solution.

Chapter 2

Requirements analysis

In this chapter we will describe the several requirements of the solution, which guarantee the development of an innovative web platform that we aim to achieve. The platform allows to resolve the current problems of the sepsis treatment process and provides new advantages to the users. Finally, we will also show and describe the mockups of the platform.

2.1. Features and system boundaries

As general design principles we can already identify the following system characteristics and features:

- Communication protocol (security, privacy and standardization);
- Authentication system;
- Secure data exchange (web-services exchange data over SSL[1]);
- Event-driven (asynchronous communication);
- Scalability (capacity to serve different number of devices in different time);
- Resilient (capacity to recover of fails instantaneously);
- Responsive (automatically synchronize page content when occur modifications in the system);
- Collect information in a centralized knowledge database;

[1] SSL – Secure sockets layer

2.2. Stakeholders

The solution involves several users during the process workflow. Users interact with the system in specific phases of the workflow. Each user assumes different roles in the system according each specific step.

All the responsibilities of each stakeholder involved in this project were established during the project kick off meeting. These roles are detailed on Table 1.

Table 1 – List of Stakeholders

Stakeholder	Description	Responsibilities
Physician	User that initiates and finishes the patient workflow in the HemoSpec platform, with direct interaction with patients.	Starts a patient's process, enabling the technician to do a blood analysis workflow. Afterwards, he is notified about the generated results, analyses patient's status and defines the right treatment.
Technician	Member that forwards the patient's workflow after receiving an order from the physician. He also manages the patient's blood analysis workflow.	Ensures that the correct blood sample is properly inserted in the machine, and verifies if all is correct to start the analysis process. Then, he has to validate the results, and allow the workflow to continue.
HemoSpec System	Catches the checked results and compare them with specific values according to defined criteria. Concludes the adequate treatment and notifies the physician.	After receiving the validation from the technician, it initiates the last workflow step. Processes the patient's data and, based in specific values, classifies the patient's disease severity. Finally, according to specific values, it suggests the adequate treatment.
Patient	Main target of the process, but doesn´t interact with the system.	

Administrator	User with maximum privileges in the system. He can manage and control all aspects of the platform.	Manages user permissions in the system: change roles or edit users' data. Can also use all permissions (physician, technician …) to manage the system and guarantee that all the data are accurate and ensure the system works correctly.

2.3. Stakeholders requirements

In this section we will describe the needs that every user has with the system. Some of the needs are specific of each stakeholder, but also exist needs that are common between stakeholders. On the one hand, physician and technician have in common the account and access to a private workspace requirement, on the other hand, they have specific requirements applied in your specific context. While physician has requirements according to the creation, starting and management of the patient's workflow, the technician has requirements related with the blood analysis workflow. The HemoSpec system has specific requirements according to the application in terms of performance, communication protocol, etc. Finally, the administrator has requirements related with management of users and system.

During the project's kick off meeting, we started analyzing the general system requirements and also the specific requirements to each stakeholder (Table 2).

Table 2 – List of stakeholder's requirements

	Requirement	Description
Physician	Access to a private workspace	See all ongoing patients' workflows.
	Create and start patient's workflow	Physician creates the patient's process in the system, workflow that has a sequence of steps and starts the workflow.
	Manage patient's workflow	Physician can consult the patient's information in the workflow and submit the right treatment.

	See workflow's state	Physician can consult the state of a processing workflow.
	Account	Full control of the account.
Technician	Access to a private workspace	See all running blood samples workflows.
	Start blood sample workflow	After receiving a notification from a physician, technician collects a blood sample and inserts it in the device, receiving a validation from the HemoSpec device.
	Manage blood samples workflow	Technician can see and manage each process in execution and, when a process is concluded, he verifies if the results are correct.
	Account	Full control of the account.
HemoSpec System	Authentication	The system has to limit authorized user access and ensure the attribution of correct permissions.
	Performance	System guarantees an efficient and fast process, avoiding any delays in the system.
	Notification mechanism	The system has a notification mechanism to inform the users that a specific activity has started.
	Communication protocol	It supports a communication protocol and allows the technician to control the blood sample workflow through the HemoSpec platform.
	Support	The system should support different web browsers.
Administrator	Manage users	Administrator can add, remove and control each user's access and permissions.
	Manage system	Administrator has access to all system's components. He can view or edit all data, in order to avoid incorrect information on the workflow. This ensures that problems or conflicts on the system don't occur.

2.4. Functional requirements

The functional requirements were defined and detailed in 4 different areas:

- System requirements: overall system features;
- Patient workflow management: overall patient workflow features;
- HemoSpec process: requirements associated with HemoSpec device and system;
- Platform management: requirements associated with the platform administration.

The first area, system requirements, defines functional requirements associated with the authentication process, the management of user accounts, patients, and consultations:

- Support different user profiles, such as physician, technician and administrator;
- Private account management: users need to be registered and have the account validated to use the system;
- Users can view and manage all patient's process;
- Users should be able to search for several consultations in the platform;
- Users should be able to search for patients;
- Technicians should be able to interact with device.

Additionally, users should be able to edit their account information, such as name, password and address.

Now, we will describe the management area of patients' workflow, which is the main requirement of the system. This allows users to have total control about the patient workflow:

- The system will be used to create and start patient's workflows and to display the current status of the patient's workflow;
- User should be able to control the actual patient's workflow;
- User should be able to analyze patient's results;
- User should be able to consult the patient's history.

On the other way, HemoSpec process is responsible for controlling all the blood analysis process. This is the most important phase of the patient's workflow, since all the decisions taken by the physician and technician depend of analysis results':

- Technicians should be able to control the analysis process (start, stop, reset);
- Users should be able to consult the analysis process status;
- HemoSpec system collects results from several modules and analyze this results;
- HemoSpec system classify results and conclude the patient's state;
- Technicians should be able to validate the results.

The last point, platform management represents several requirements that are fundamental to ensure that all work correctly. An administrator should be able to configure data associated with specifics organizations, patients, etc. Following, there are requirements related with platform management:

- Administrator should be able to manage several organizations;
- Administrator should be able to add, edit and remove roles associated to several organizations;
- Administrator should be able to manage users;
- Administrator should be able to add, edit and remove patients;
- Administrator should be able to manage devices associated with each organization.

2.5. Non-functional requirements

In addition of all functional requirements listed in the previous section, there are other issues and restrictions that our solution needs to overcome. Below, we will describe the main non-functional requirements of our system that are responsible to cover these issues and restrictions.

2.5.1. Usability

Usability and user-friendly interfaces should be an important requirement for a complete platform. These characteristics ensure a fast and easy perception of the content presents in the web page. In this way, users instantaneously observe the most important items and can be easily guided by the system. The user interface must keep the content as concise as possible, allowing user that first see the most relevant and recent information

and after the less relevant information. The interface should present the information by sections allowing distributing the information by theme or relevance. To facilitate the web page legibility also is important an adequate spacing between sections of information or even characters, words, lines, etc. It is also important to use visual strategies like photos, charts and graphs that can easily express rich information [2].

So, HemoSpec will be a web application that follows the characteristics mentioned before and will be based on an intuitive, simple, user-friendly and direct visual interface. It intends to be easy to use, with only a few steps required to reach specific information [3].

A documented API, a tutorial and a few examples of the use of this API should be provided. HemoSpec should be in available in English, to reach a larger range of users.

2.5.2. Resilient

Actual web platforms should always be available and able to handle several users at the same time, over the same workflow, without failures or, in extreme cases, ensuring spontaneous system's recovering. It also needs to ensure that when a user executes an action, this action is executed. If the system breaks in the middle of an action, the system must ensure that user data are saved, and are available for later submission [4].

2.5.3. Performance

The visualization of the platform is affected in great deal by performance issues, such as, the loading of data and the speed of rendering the browser window. All the tasks involved in the platform should be performed the quickest as possible in order to avoid long delays that compromise the entire application usability. However, performing these tasks using standard web developing techniques, such as HTML, CSS and JavaScript, can be more difficult than it looks like. Thus, the design and optimization of the rendering and document parsing algorithms is crucial to maintain the desired performance.

2.5.4. Supportability

The system supportability is another important requirement, allowing the access to the developed solution in almost all modern browsers (Chrome, Firefox, Opera), ensuring that the several users can use different browsers according to their preferences. Also, it is possible to use this solution in different operating systems according to the preferences of each user.

The platform must also guarantee that version's updates made to the system won't affect its normal performance.

2.6. Architectural requirements

After we have analyzed and described functional, non-functional and stakeholders' requirements, we need to have in consideration the architectural requirements. The architectural requirements are important because they ensure the correct working of the solution.

To comply with all requirements we have to define an appropriate architecture for the system that guarantying the working of the solution. To define these requirements we need to consider the software, hardware and communication interfaces, as well as the type of required communications.

2.6.1. Software interfaces

Software interfaces are developed under a web server, which is responsible for receiving a request, process this request and return the answer.

So, to ensure a correct communication between the client-side and the server-side of the solution, it is fundamental that we have appropriated interfaces that manage all requests, deal with possible errors and return the correct answer to the client.

Summarizing, the web server ensures the system's communications and keeps data synchronized.

2.6.2. Hardware interfaces

As described in the last section, to support software interfaces we need a web server, but to maintain a web server running we need a specific hardware, in concrete a server that support the respective web server as also the HemoSpec database.

2.6.3. Communication interfaces

After we have considered the software and hardware interfaces, we also need to take in attention the communication interfaces. Since the communication interfaces are the base to ensure the correct communication between the client-side and the server-side of the HemoSpec solution. So, the system requires a protocol to ensure the communication between all components involved in the system. The respective protocol is based in a standard API that respects a specific communication's structure. This API will use the RESTful web services, to ensure a correct and easy communication.

2.6.4. System Communication type

Considering that the HemoSpec system has several actions running in real-time and the system involves more than a unique user, it is necessary to apply asynchronous communication in order to allow processing multiple messages in parallel, without blocking.

This type of communication avoids queuing data, guaranteeing that more data are transferred at the same period of time and also it ensures that the communication is faster compared to the applications that use synchronous communications. Therefore, the most important characteristic of this communication type is assuring non-blocking parallel messages, permitting instantaneous data exchanges.

2.7. User interface requirements

User interface design has the goal of making user's interaction as efficient and simple as possible, ensuring that the user can perform an action with the minimum number of clicks. It is important that the interface will be simple and agreeable, so that, the user instantaneously visualizes the relevant information and does not lose time searching the information in the middle of the context.

To guarantee that a user-friendly interface it is necessary to follow a list of specific requirements:

- User interface is adapted to user experience and knowledge;
- User interface is adequate for each task;
- Providing control to the user about the interface;
- User interface is easy to use and quick to learn;
- The interface should be redundant, so that ensure an easy way to doing a specific action.

Summarizing, the Platform's design will be taken into account, so that, the user interface must be attractive and clear. It should allow users to easily interact with the system and also should be provided the necessary tutorials.

2.7.1. Layout and Navigation Requirements

- Dashboard/workspace where users can find their studies;
- Dashboard where users can manage the devices;
- A screen to create and manage the workflows;
- A screen to manage personal data;
- An administration screen to manage, create or delete organizations, users, patients, devices and control added information;
- A screen with FAQ and tutorials about the platform.

2.7.2. Consistency

Consistency is increasingly an important requirement in the websites constructions, because nowadays we can access to a specific website from

several devices (tablets, smartphone, smartTV, LCD, etc.). So, it is important to keep a consistent website to ensure that the users do not notice a difference using different screen layout and do not have to learn new tricks to navigate and manage the website due the size. The user should feel comfortable to use the website, so he continues to use it.

On the other hand, interface should have a uniformed look. The uniformed look should follow a set of different types of consistency, such as:

- graphic consistency;
- icon and button consistency;
- color consistency;
- typographic consistency.

2.8. Interactions requirements

In this section we will present and explain all relations that occur between system's components during the process, where we detail in specific each relation and finally we will present all interactions that occur during all workflows and that involve all components.

2.8.1. Relation between system's components

Figure 3 is a scheme that describes an overall HemoSpec architecture and details the several architectural components that comprises the system. In general, the system is divided into three different assets: 1) the HemoSpec platform, which is the most important component; 2) the HemoSpec device; and 3) the Client-side interface that is a sub-component of the HemoSpec platform, although we consider here as an individual component.

We will start describing the relations of the Client-side interface. The interface is fundamental to allow the user interacts with the platform, the interface allows that different types of users interact directly with the system and each user type has appropriated access rights. Both physicians and technicians can work in several cases, where physicians manage more in general the patient's workflow, whereas technicians manage more in

particular the blood analysis workflow. During the patient's workflow which is part the blood analysis process, every change that occurs is instantaneously synchronized in the platform. This allows that each user having instant access to the updated information.

HemoSpec device is the component responsible for control blood analysis process. The device is formed by 4 different modules, each module is responsible for doing a specific part of the blood analysis process. After physician starts a patient workflow through the Client-side interface, instantaneously the technician is advertised to execute the blood analysis. So the technician starts the analysis also through the Client-Side interface and after the analysis is terminated, he validates the data or restarts the process in the case of any anomaly in the process. If the data is validated, the platform stores this data to accelerate the patient's workflow and process the final step of the process.

Finally, we will describe the relation of the HemoSpec platform. This component is the main part of all process, because coordinates all relations between the Client-side interface and the HemoSpec device. And manage, control and processes all system's workflows.

Summarizing, HemoSpec platform coordinates all the workflow, synchronizing all performed tasks and manage all the data-mining process to conclude the patient status. After obtained the actual patient status presents this information to the final users.

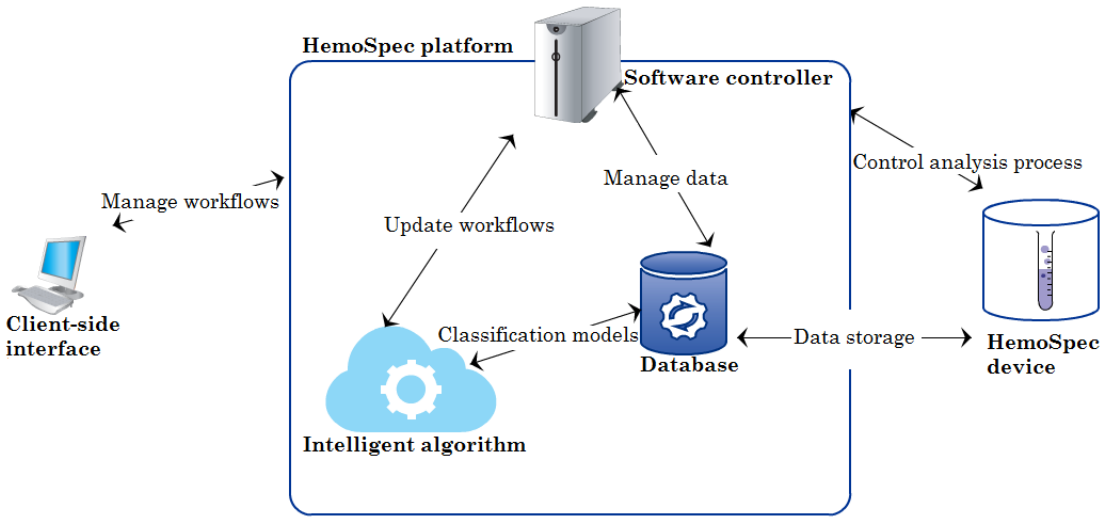

Figure 3 – Architecture of the HemoSpec Platform components

2.8.2. Interactions between components

In this section, we present all the interactions between different components, divided into three phases.

Figure 4 represents the first phase of the patient's treatment workflow. Basically, this is initiated by the first interaction between the patient and the physician, while in parallel is recovered the patient' blood that will be used in the analysis process. In the final of the patient-physician interaction, physician creates and starts the patient' workflow. After, this information is submitted to the system, the system will automatically notify the technician that a new process has initiated. The technician receives information about the process, catches the respective blood sample and introduces it into the device. After he inserts the sample into the HemoSpec device, the device notifies the technician if all is well with the placement of the sample or if not.

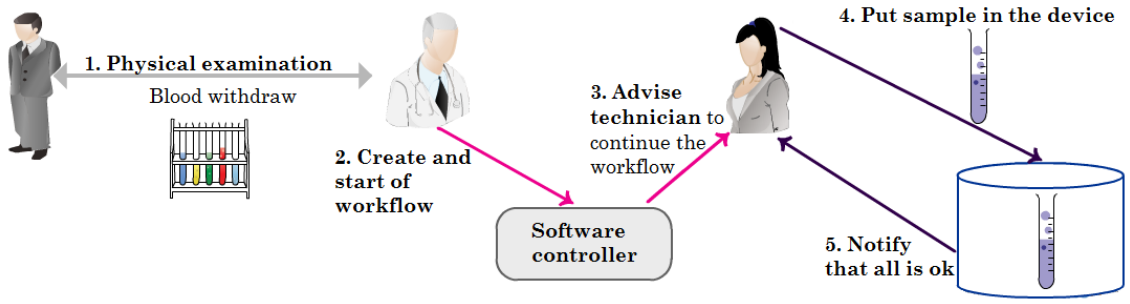

Figure 4 – System's architecture before the analysis workflow

In the second phase the technician starts the analysis process, through the web interface. Since started the analysis process, this action will be controlled by the HemoSpec platform.

The analysis process requires only a minimal amount of patient's blood. During the analysis, the HemoSpec device will perform a hemogram count and a detailed Raman spectroscopic characterization of the leukocytes.

During this process, users can verify the status of each analysis module in real-time and when each module returns a status value "Completed". After the device concludes the parallel analysis process and the software controller

(platform) gathers the results of each specific module, data are stored in the platform database.

Finally, the results are delivered to the technician, who validates them, or in case of any anomaly, restarts the process (Figure 5).

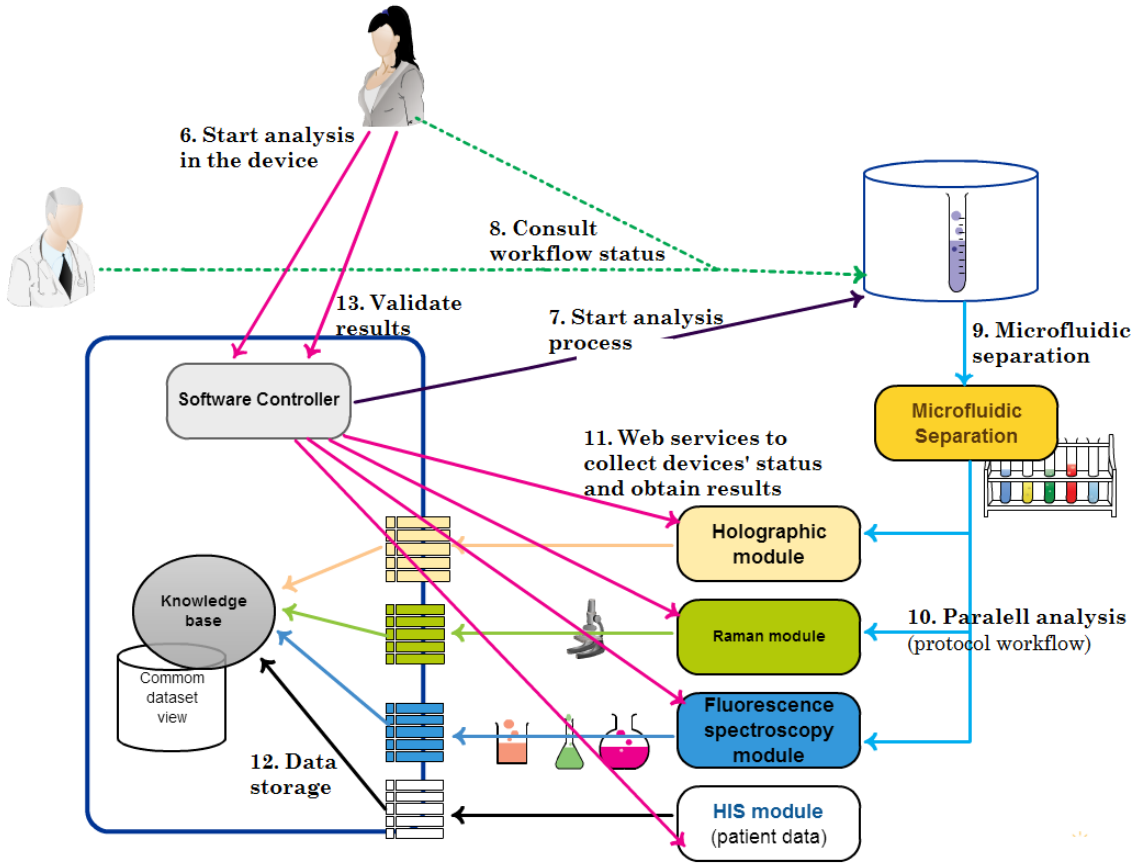

Figure 5 – System's architecture during the analysis workflow

The third and final step of the patient's treatment workflow is detailed on Figure 6. After the results validation, the system starts a data mining process, where it firstly occurs the data fusion from the several modules. The next step is to make a classification models based in standard values that classify the severity of patient status and notify the physician about the patient status.

The information provided to the physician and justified according with the referred values ensures the liability of the classification process.

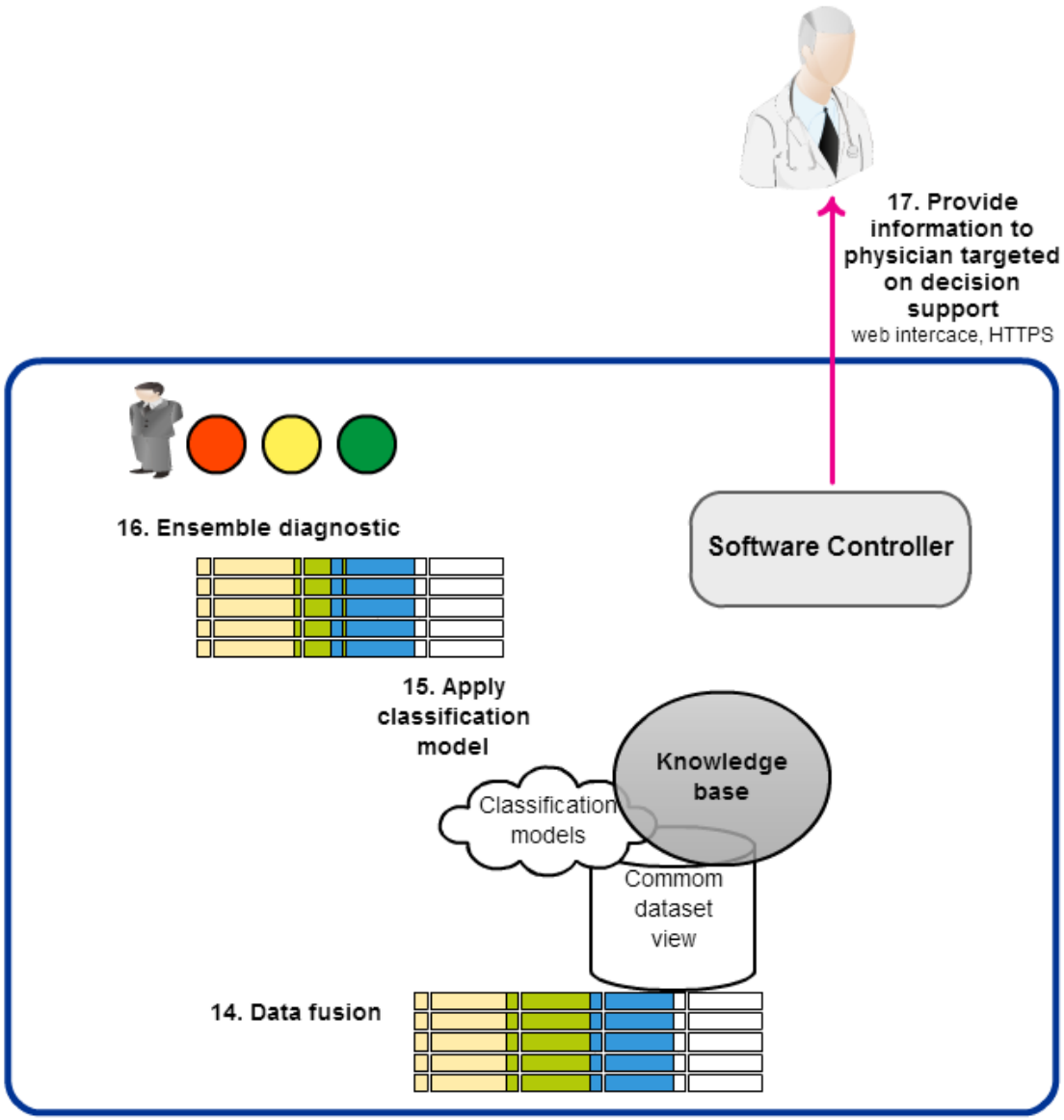

Figure 6 - System's architecture after the analysis workflow

2.9. Application requirements

To finalize the requirements analysis of all the solution, in this section we will show and explain all the use cases related with the solution and that express all possible sequences of interactions between system and users. We will also show use-case diagrams that present all available functionalities of the system and we will present an activity diagram that explains the general workflow [5].

2.9.1. Use cases

In this sub-section, we will present all the existent use cases that represent all the interactions between actors and the system [6]. We divided the several use cases by different packages, where each package describes use cases related with specific sections of the solution. This way, the solution was divided in five packages (Table 3)

Table 3 – List of use cases

Package	Use case	Use Case Description
Authentication	Login	The system verifies registration and permissions.
	Logout	User terminates his session in the system.
	Recover password	The user forgets his password and asks for a new one.
User	Invite	Administrator sends an invite.
	Register Invite	Member receives an invite and finishes the registry.
	Change role	Administrator manages the users' access and permissions.
	Search	Administrator searches for a specific member.
	Edit	User can edit specific data.
Patient workflow	Create	Physician creates a patient process in the system.
	Start	Physician begins the patient's workflow.
	Search	User searches for a specific patient's workflow.
	View Status	User observes the patient's workflow status.

	View results		User analyses the patient's workflow result.
	History		Physician and Technician can consult and analyze past patient's workflows.
HemoSpec device	Clean		Technician prepares the device to start another blood analysis workflow.
	Start		Technician initiates the analysis process.
	Stop		Technician stops the analysis process.
	Status		Technician consults the analysis status.
	Collect results		HemoSpec joins results from all modules.
	View results		Shows the analysis report.
	Validate results		Technician analyses and indicates if the results are accurate.
	Restart		Technician can reinitiate the analysis process.
HemoSpec system	Data fusion		System joins all patients' data.
	Data storage		System storages the patients' data in the database.
	Classification		According to the analyzed data, the patient's condition is concluded. This information provided enables a better sub classification of the patient groups and improves the diagnostic accuracy.
	Treatment		System notifies that the patient's workflow is ready.
	Notification	Create	System updates the workflow status and creates a notification.
		Add	After the notification, the system notifies the next user involved.

	View	User receives a notification and can view its content.
	List	System shows a list of all pending notifications.

2.9.2. Use- case diagrams

In this subsection we will describe in detail the set of packages referred in the last sub-section and we will also specify the respective use cases [7].

So, we will begin with the authentication package that is presented in the Figure 7. The figure represents a common package for all users. Like we referred, the "Authentication" package includes use cases associated with system access. Follow, we will describe the list of use cases:

- Login – represents the interaction between the user and the system, where the system verifies registration and permissions of the user to access;
- Logout – it is the inverse case of login use case, where the system ends the session of the user;
- Recover password – This use case represents a specific case, when the user forgets his password and asks for a new one to access again.

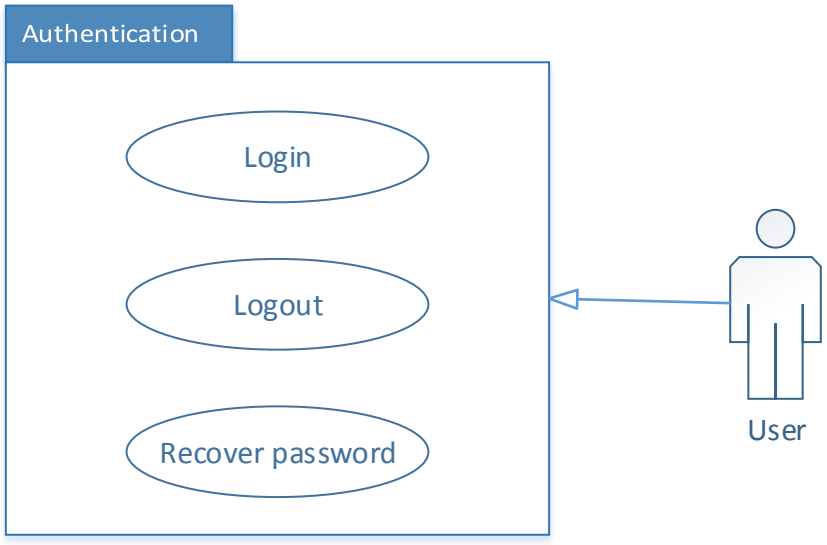

Figure 7 - Use case diagram for the Authentication package

Bellow, we will present the User package that involves two different user types. On the one hand, we have the system administrator who has access to all use cases, on the other hand, we have in particular the physician and the technician which only have access to the Register invite and Edit use case. We will describe all the use cases of the User package showed in the Figure 8:

- Invite – when the administrator wants to add a new user to the system, he inserts user data in the system and automatically the system will be sent an invite to him;
- Register invite – a member receives an invite from the system and finishes the registry process;
- Change role – administrator can edit the user's access and permissions;
- Search – use case that represents the action of the administrator searches for a specific member;
- Edit – Each user can edit your specific data.

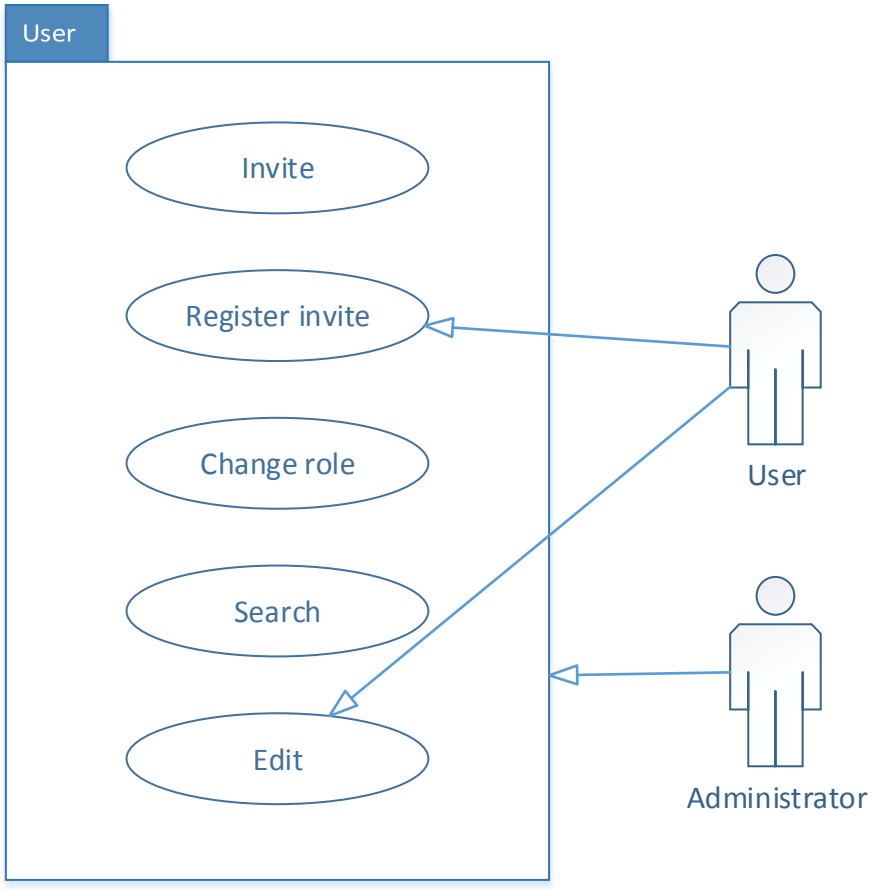

Figure 8 - Use case diagram for the User package

Follow, we will speak about the Patient package that includes all uses cases that directly interacts with the patient's workflow during the processing time. The stakeholders involved in this package are the physicians and technicians. Now we will describe the several use cases that belong to the Patient package (Figure 9).

- Create – physician creates a new patient process;
- Start – physician begins the patient's workflow;
- Search – user searches for a specific patient's workflow;
- View Status – user observes the patient's workflow status;
- View results – user analyses the patient's workflow result;
- History – Both, the physician and the technician can consult and analyses past patient's workflows.

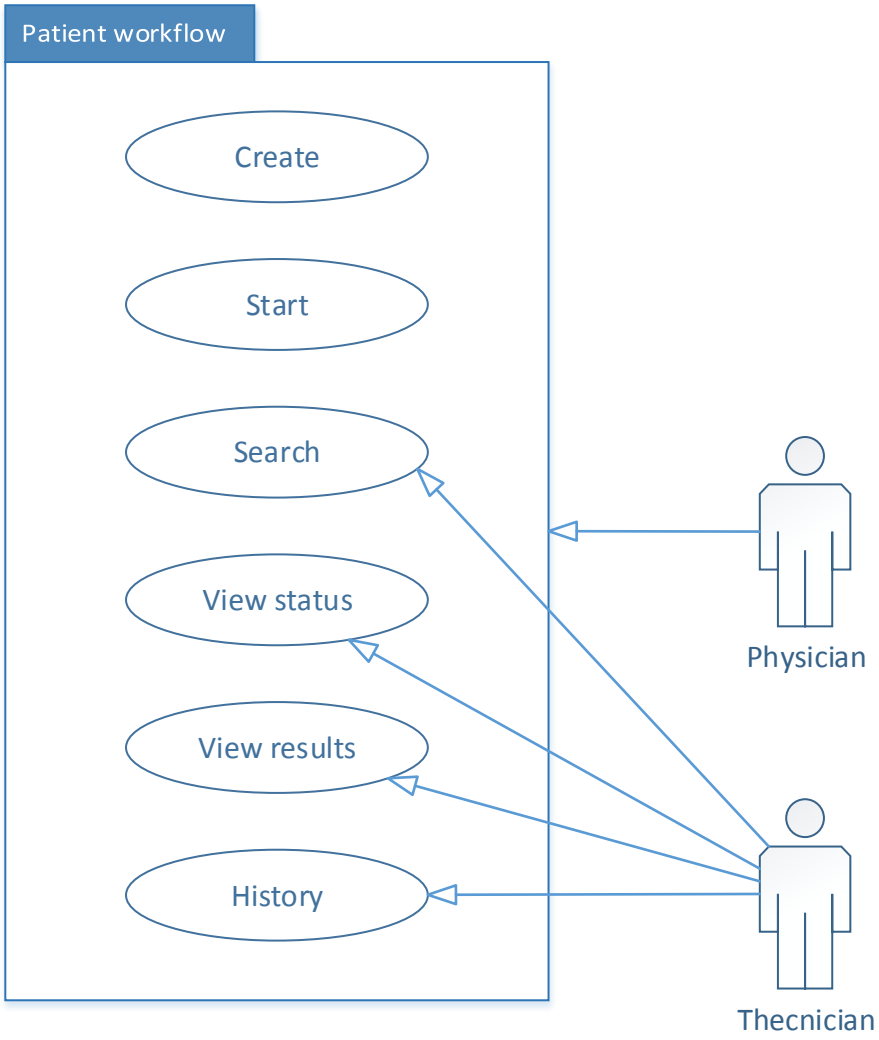

Figure 9 - Use case diagram for the Patient Package

Now we will describe the HemoSpec device package that represents all use cases that have a direct relation with the device. Two different users interact with the device and they have specific functions. The main user in this package is the technician who has permission to interact with major of the use cases, whereas the HemoSpec system can only interact with the View results and collect results use cases. After we can see the list of all these specifics use cases:

- Clean – technician cleans the device and allows preparing the device to another blood analysis process;
- Start - technician begins the analysis process;
- Stop – technician can break the analysis process;
- Status – technician can visualize the analysis status;
- Collect results – After terminates the blood analysis process, the HemoSpec system joins results from all modules;
- View results – shows the analysis report;
- Validate results – after terminate the blood analysis process, the technician analyses and validates the results if the results are accurate;
- Restart – In the case of an error or if the results are not accurate, technician can reinitiate the analyses process.

In the Figure 10, we can see a scheme that shows the package, the respective use cases and the involved users.

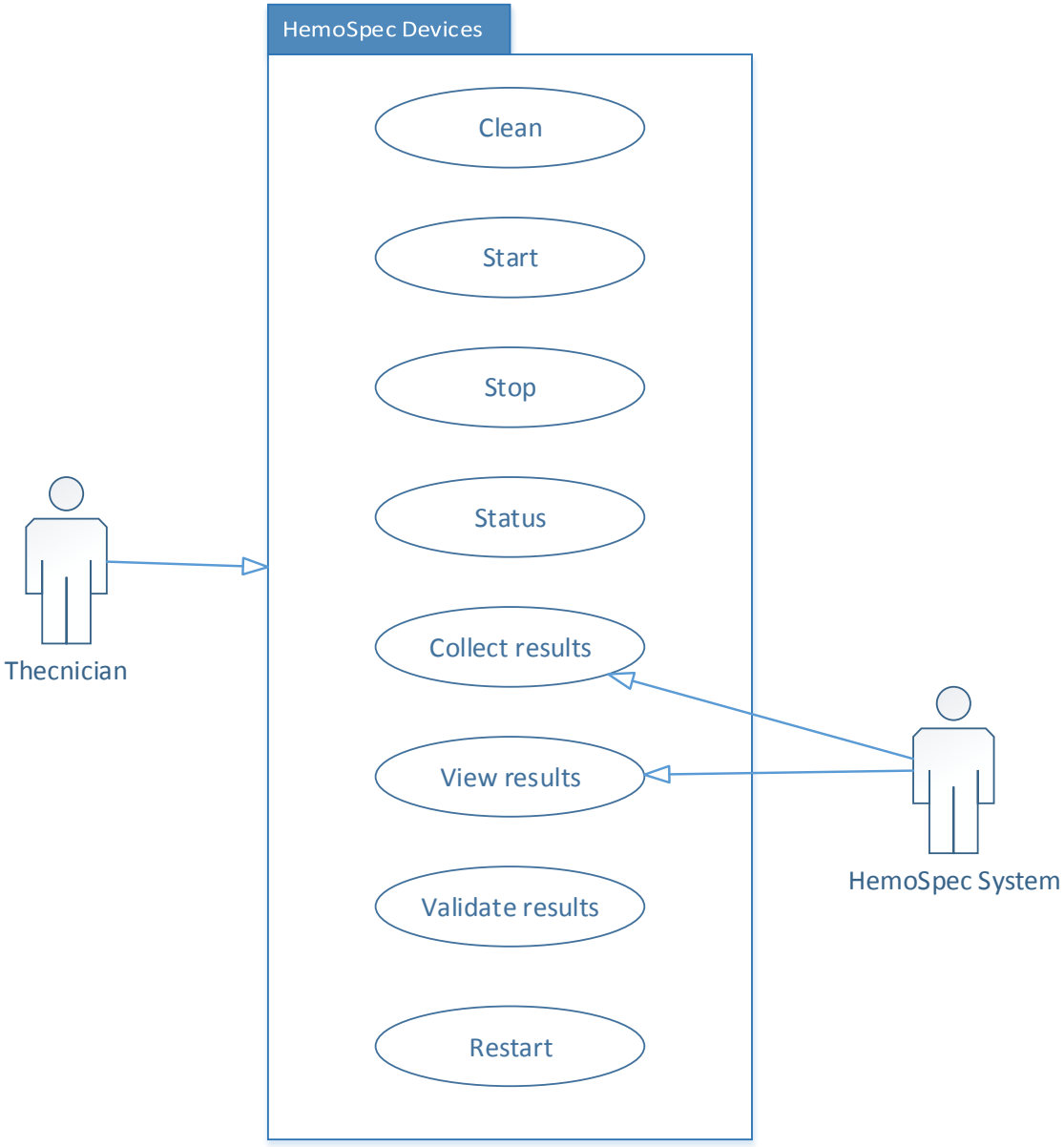

Figure 10 - Use case diagram for the HemoSpec Device Package

Finally, we will describe the HemoSpec system package that contains all use cases related with data mining and background process, which are the last step of the treatment's workflow. We present follow a list that shows all the use cases presented in the package:

- Data fusion – represents the join of all patient's data;
- Data Storage – storages of the patient's data in the database;

- Classification – use case that represents the data analysis process and the respective patient's condition classification;
- Treatment – notification that the patient's workflow is concluded.

However, inside this package exists a sub-package that represents the Notification package, where the use cases related with the communication in the system are included (Figure 11).

- Create – creates a notification and updates the workflow status;
- Add – notifies the respective user after the creation of the notification by the system;
- View – represents the visualization of the notification;
- List – list of all pending notifications.

In both packages, only the "HemoSpec system" has access to all use cases.

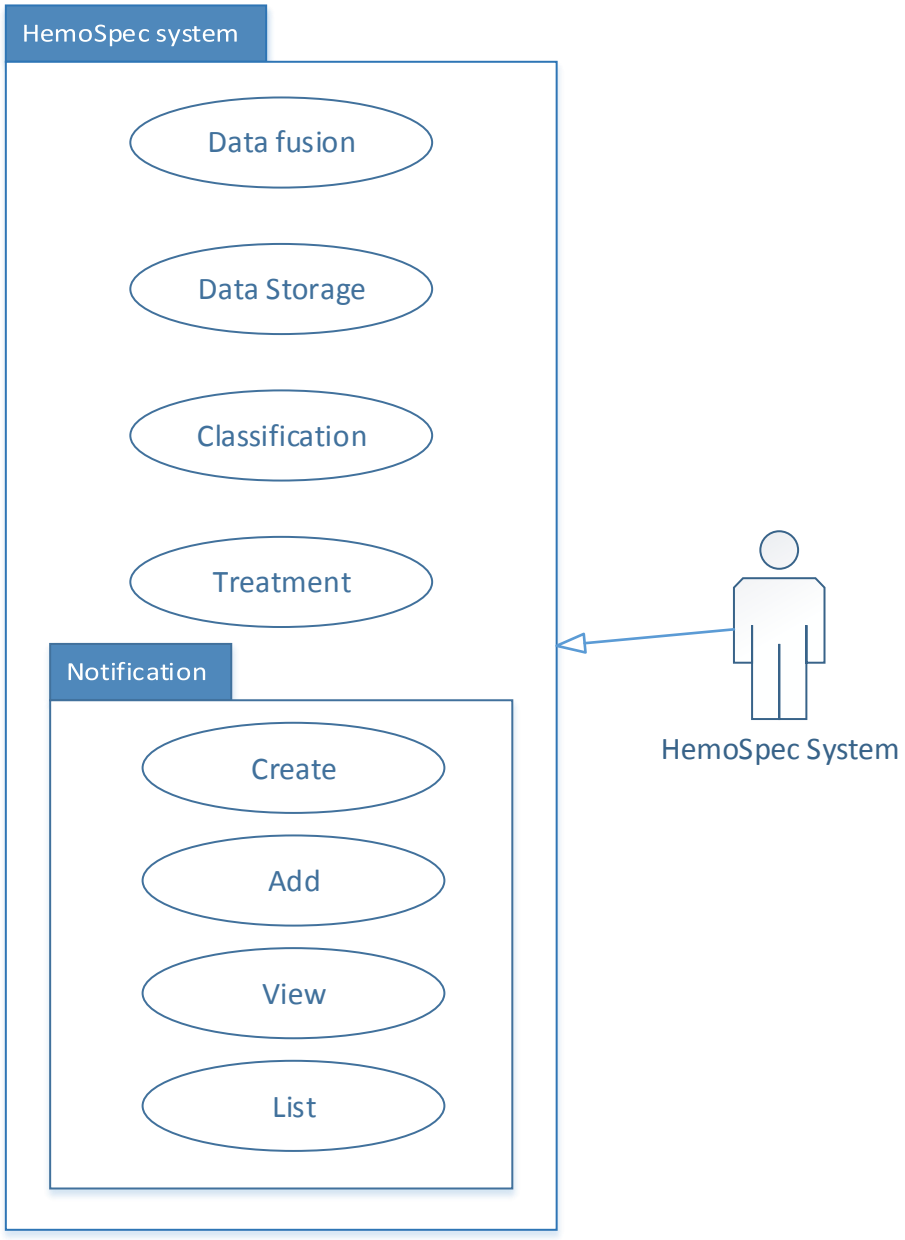

Figure 11 - Use case diagram for the HemoSpec system package

2.9.3. Activity diagram

Since we already described both the use cases and the use-case diagrams, now we will describe the activity diagram. This diagram shows a complete scheme that describes the normal system's workflow [8]. Specifically, we can notice that the diagram represented in the Figure 12, includes multiple interactions among physicians, technicians and the system.

The workflow starts with an initial interaction of the physician, he creates the patient workflow. After the patient workflow is created by the physician is necessary that the workflow continues to working and for this the physician starts the patient workflow. All these actions will be considered by the HemoSpec platform that will process respective actions and processing the workflow. So, to continue the workflow the HemoSpec platform will going to notify the technician that exist a new workflow running and that he should maintain the same. Upon receipt of the notification, the technician will clean the device for after can start the device that will do a new blood analysis process. During the analysis process all users can see the status of the analysis. After completion of the analysis process the HemoSpec platform will collect all results and present these results to the technician. When the technician analyzes the results he takes a decision if the data are correct or not. If the data are not correct executes another analysis process, else validates the results. Afterwards the validation, the HemoSpec platform going to store the data and after the storage process doing a data mining process, calling of "classification". Finally, when the results are provided to the physician, the physician only has to consult results and takes an appropriated decision to the patient condition.

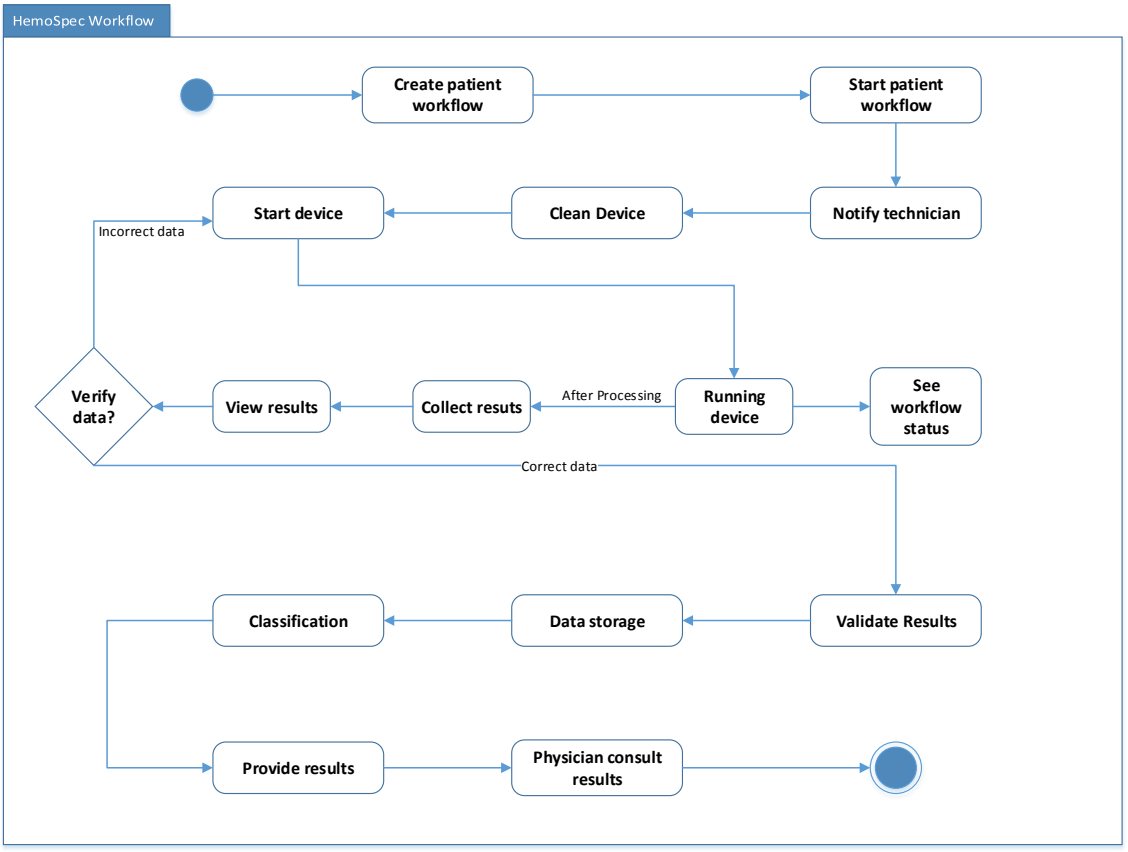

Figure 12 - Activity diagram for the HemoSpec workflow

2.10. Mockups

After analyzing all requirements, we had in consideration all information obtained and we started to draw the respective mockups.

To build mockups, we decided to use Pinegrow Web Editor[2], which is a modern editor that allows to create responsive websites in a faster way. For each page we can incorporate specific frameworks like Bootstrap [3], Foundation[4], etc., which turns the interface appellative. We can create a website only doing drag and drop of the elements, still having access, at the same time, to the generated code.

During the mockups development, we defined initially three main pages. The first page is the login page that ensures a secure access to the platform,

[2] http://pinegrow.com
[3] http://getbootstrap.com
[4] http://foundation.zurb.com

after login page, we divide the platform in another two pages, patient page and device page like we can see in the figure below.

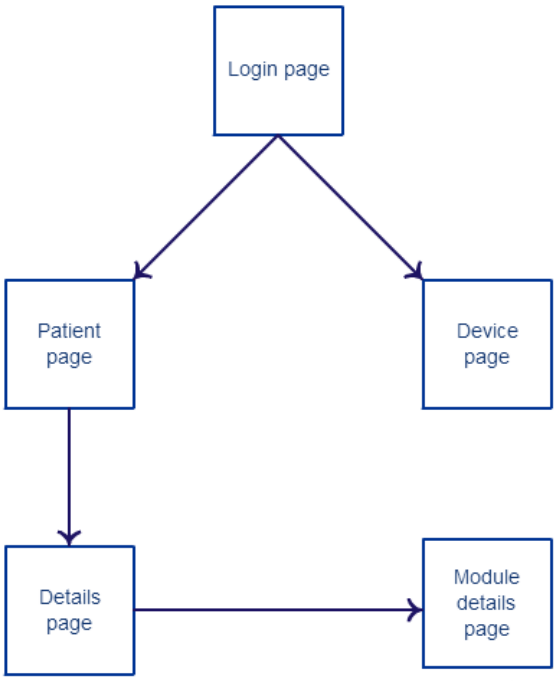

Figure 13 - Scheme of mockups structure

How we can see in the Figure 13, patient page still has more two pages associated. The details page shows all information related with a specific patient's consult and presents data obtained instantaneously during the analysis process. The module details page shows information related with a specific analysis module where technician can add comments related with this specific module results. Below we will explain and show all mockups pages.

So, we will start to show the login page (Figure 14) that validates the user data access. If user data are validated, he will be redirect to the patient page. Otherwise the same page will be reloaded.

Figure 14 – Login page

After correct authentication, the user has access immediately to the patient page, where he can observe all patients under study. In the Figure 15, we present the several states that a consultation can have during all the patient's workflow.

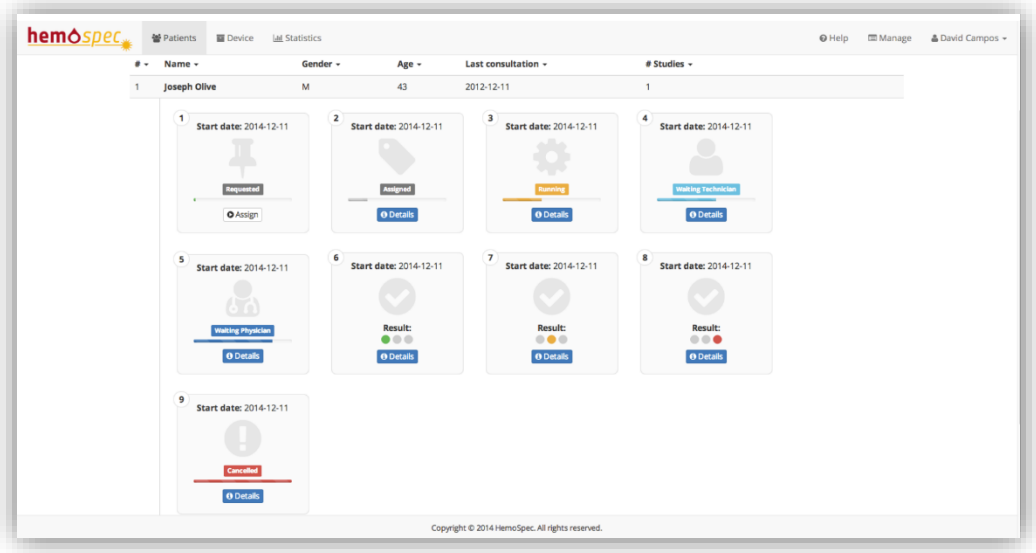

Figure 15 - Patient page, where we can see the several diagnostic phases

A consultation is automatically associated one respective analysis, which provides information about the entire analysis process. To visualize this information, users access to the details page and observe the current state of each analysis module. In Figure 16, we visualize that 2 modules have already terminated its analysis, but the analysis process continue running because there still are modules under processing.

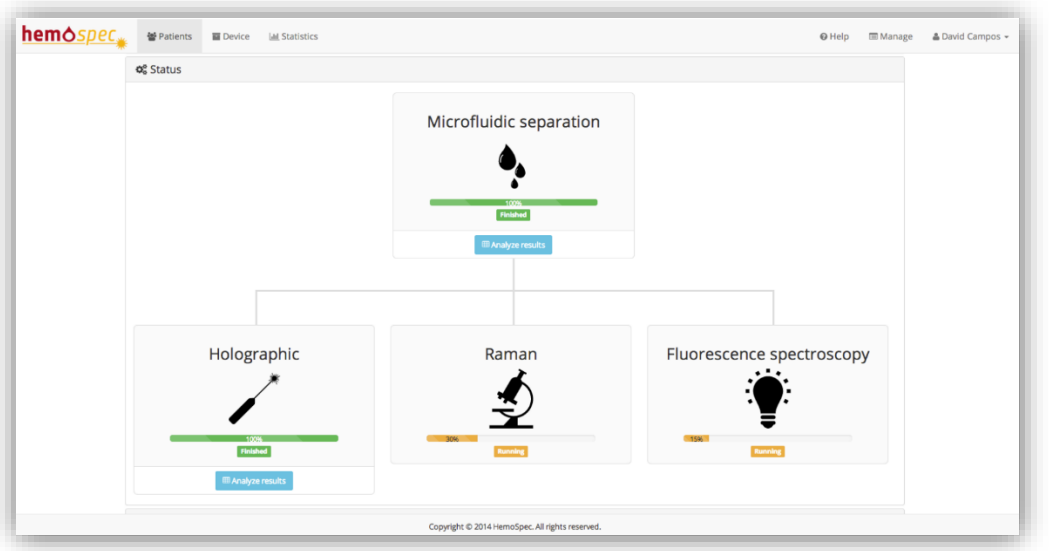

Figure 16 - Details page that shows all analysis information associated with a specific consult

Like we see in the figure above the analysis continuing running. Whereby each user can consult the module's details page associated with the terminated modules and visualize the information of the respective modules, represented in Figure 17.

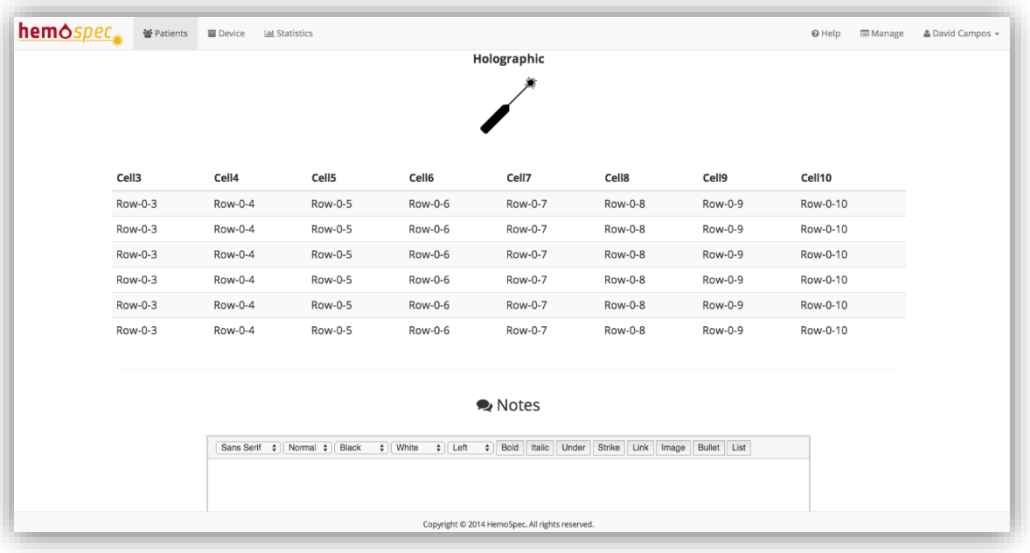

Figure 17 – Page that expresses information of a specific module associated with a specific consult

After all modules terminate the analysis, HemoSpec system saves the results in the database, allowing technicians to consult these data through HemoSpec platform. It still permits the technicians to take a decision accordingly to the results. The technician will validate results if all is correct or he/she will re-initialize the analysis process in the case of any specific parameter or result is incorrect (Figure 18).

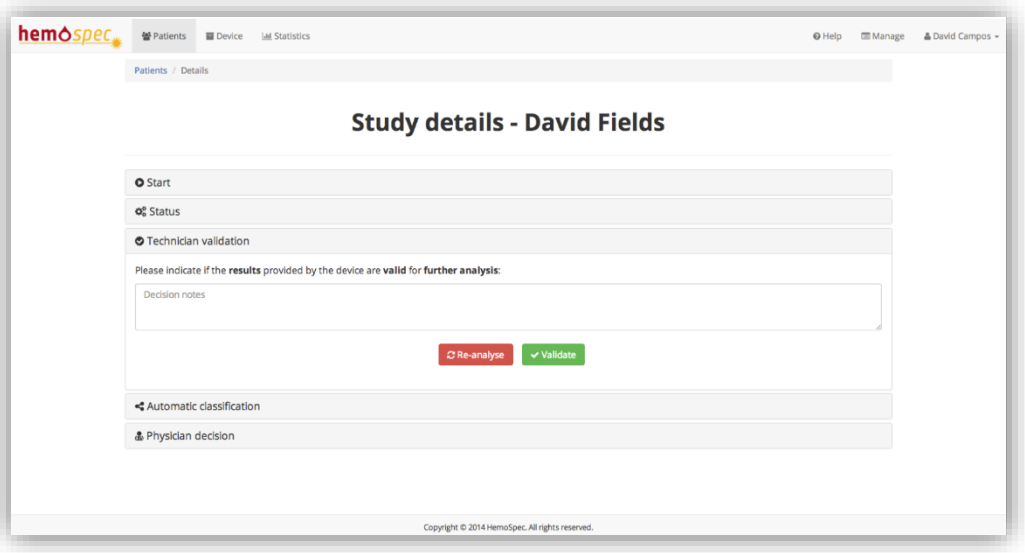

Figure 18 – Section page when technician has to validate data or reinitialize the analyze process

Once that technician validates results, the platform shows immediately the page that presents the automatic classification results to the physician (Figure 19). After analyzing these results, the physician submits an appropriated decision to this case.

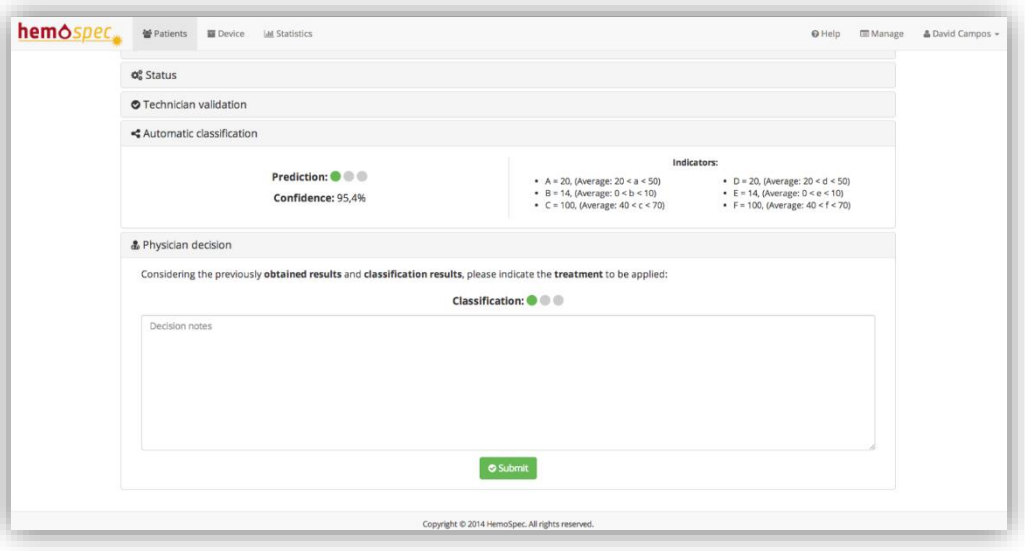

Figure 19 – Section page when physician see the final results and can take a decision

During all the analysis process, users can also visualize and explore the device page, where they can see the current state of each module and can consult the historic associated (Figure 20).

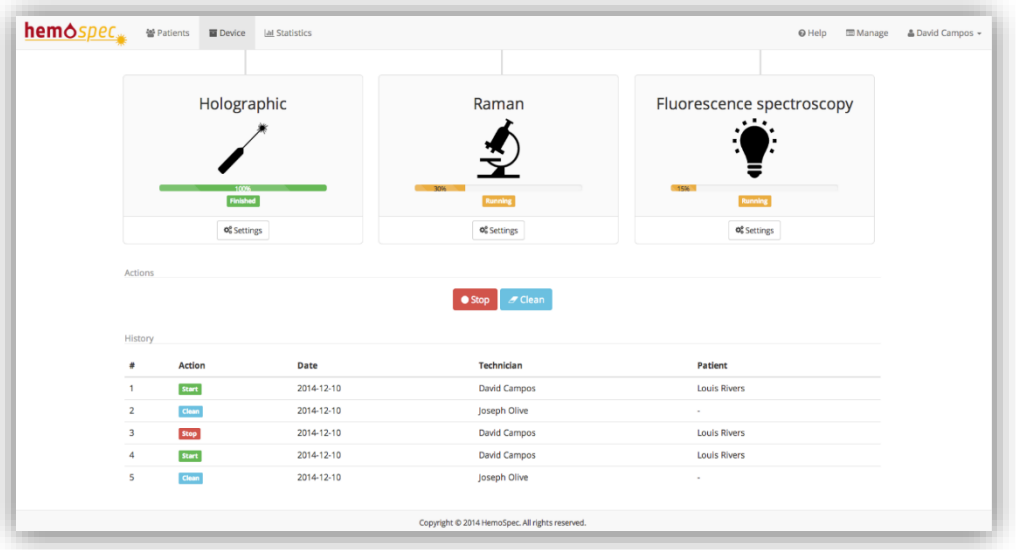

Figure 20 - Device page that shows all information related to device

Chapter 3

Architecture Proposal

Based on the requirements presented in the last chapter, we will propose an architecture that complies with all features required by the project. With the objective of building a fast and natural platform that respects all requirements and ensures a user-friendly user interface, we need to have in consideration the chosen technologies that will be responsible for the core of the application.

3.1. Overview

As already mentioned, our solution is based on a web platform, composed by the client and server side.

On the one hand the client side is the part that represents the user interface through which each end-user can interact directly using their web browsers. On the other hand, the server side ensures the management and data storages in the database. We use the MySQL[5] database for managing and interacting with data stored in the system. The server side also ensures a communication protocol.

In the Figure 21 we present a general overview of the architecture. During all communication between client and server side the exchange of data is made using the HTTPS protocol, which ensures the confidentiality, anonymity and security of data using authenticated and authorized services, avoiding its use by third parties [9].

[5] http://www.mysql.com

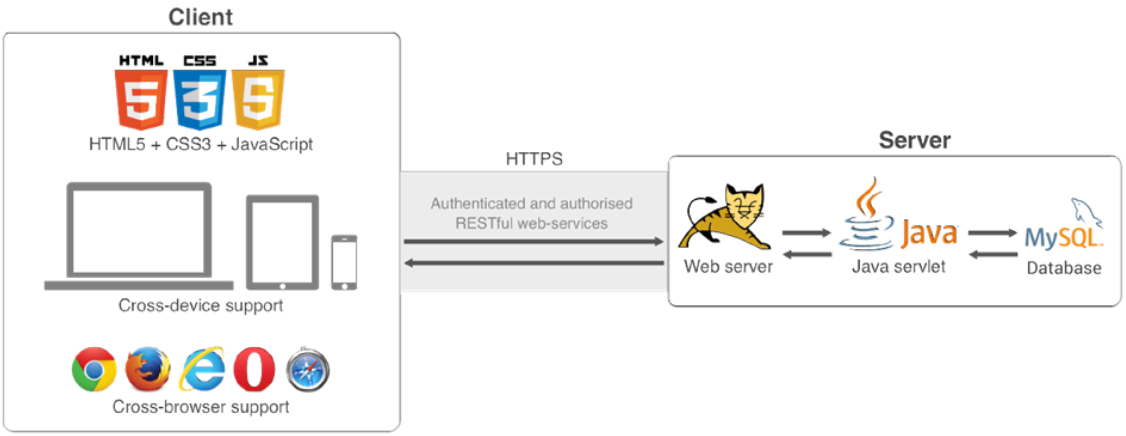

Figure 21 - General overview of the HemoSpec Platform architecture

3.2. Client

When building a web-based application, two of the majors concerns focus on the usability and user-interface. On the one hand, we must ensure that when we will develop the platform, this has to be more user-friendly as possible. On the other hand, performance issues, such as slow data representation, should be taken in consideration, since they could compromise the usability of the solution.

To resolve all these issues and limitations our solution provides an innovative and flexible platform, since it is easily available for most devices with an internet connection. On the other hand, performance and supportability are key-factors that we must be aware of while developing our application. Thus, we choose standard web technologies, making HemoSpec Platform supported by the most widely used web-browsers on the market.

This way, in our platform we use $HTML_5$ that will structure and present the content in the page and we use CSS_3 to format and give visual meaning to the content. To ensure a good interaction with user interface, we use JavaScript that will control the dynamic content, ensuring an asynchronously communication with the available services and after the invocation of these services update the platform content.

These technologies allow us to create a widely supported platform, however, it is very important to assure that the same standards work in different browsers. Nonetheless, even though these technologies deliver fast

representation of information and cross-browser support, performance issues can emerge with large amounts of data. Thus, the development and implementation of fast and optimized algorithms are crucial to maintain a fluid application and enable fast load and visualization of data. Moreover, a web platform which includes all the requirements listed in the previous chapter can achieve a few thousands of lines of code on both server and client sides. Likewise, if we are building a web platform using JavaScript – like HemoSpec – it is extremely important to use an architectural pattern that add structure to all the data and facilitate the management of the code.

3.2.1. JavaScript

Since developers began to use JavaScript, most of the interactions between users and web platforms are performed faster. An important JavaScript fact is that tasks are executed faster, because those tasks are processed and completed almost instantaneously on the client side what avoids processing data on the server-side and sending again to the client side. This situation avoids consuming local, as well as server, bandwidth and time.

Traditionally, web applications leave the heavy-lifting of data to servers that push HTML to the browser in complete page loads. This way, the use of client-side scripts was limited trying to improve the user experience. Nowadays, this relationship has been inverted, client applications pull raw data from the server and then render it into the browser when and where it is needed.

In relation of the asynchronously communication until recently the most used forms to control asynchronous operations was through callbacks. The big problem of using callbacks is that when we have to chain multiple callbacks the code stays very hard to follow and understand. The solution used for us are promises that avoids the callbacks problems [12]. After we compare some libraries [13], we choose the bluebird library [14], since in general is the best. We present in the Figure 22 the comparison of the last versions on the Chrome browser, but you can consult comparison of all versions in the link.

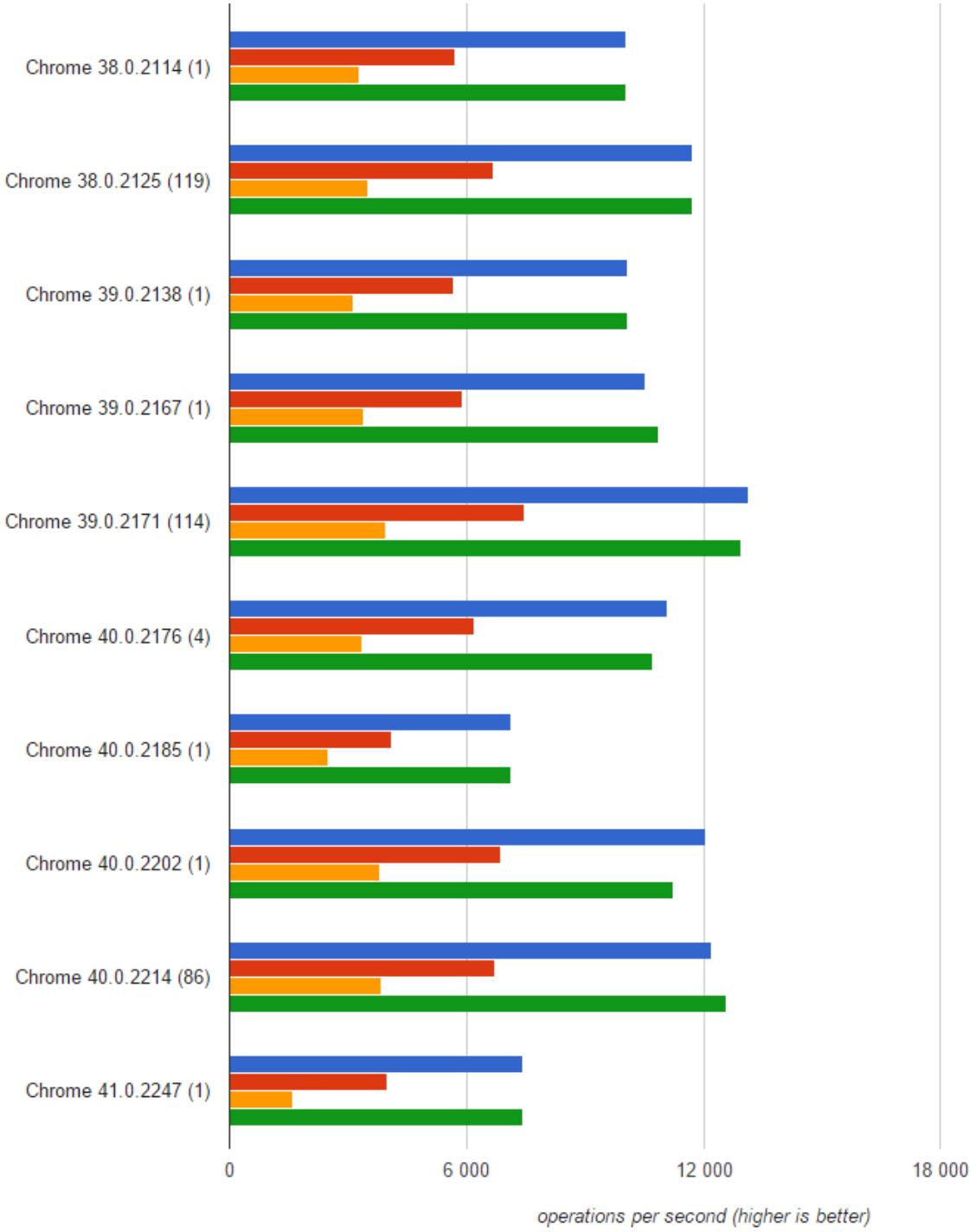

Figure 22 - Comparisons between several promises libraries (blue – Bluebird, red – Pimp, orange – Q and green - RSVP)

To manage all code that produces the platform and that complies with all requirements listed in last chapter, we adopted a Model-View-Controller (MVC) architectural pattern that simplifies the structure and management of the code [15], [16]. This pattern divides the application in three parts:

- Models, which represent the knowledge and data in an application, like an Organization, a patient, etc. Models advise other parts of the application when their state changes.
- Views, representing the user interface of the application. The views update the user interface when occurs a change in the models.
- Controllers, which handle events that occur in the view (e.g., user actions) and invoke the respective update in Models.

In our solution, we decided to implement RequireJS[6] in order to improve the speed and quality of the code [17]. RequireJS is a modular script loader that avoids repeating the "includes" scripts in each HTML file. For example, if we have 2 pages and we need to import jQuery[7] and Bootstrap[8] files to both pages, we only need to create a JavaScript RequireJS file where we write each include. So, with this strategy, we only need to include the reference of the JavaScript RequireJS file created on each page, and we avoid to repeat write several times each include. Also, we integrate WebJars[9] that allows loading dynamically each necessary library to each page, what ensures a faster platform.

3.2.2. User interface

Our application is designed to be simple and easy to use, providing a comfortable user experience while helping users to use the platform. In order to create a fluid interface, we take advantage of several high-end technologies regarding to front-end developing and templating, such as:

- **jQuery:** jQuery is a fast, small and feature-rich JavaScript framework designed to simplify the client side scripting of HTML. It makes things like HTML document traversal and manipulation, event handling, animations, etc. jQuery is behind over 70% of the

[6] http://requirejs.org
[7] https://jquery.com
[8] http://getbootstrap.com
[9] http://www.webjars.org

most visited websites[10], making it the most popular JavaScript library currently used [11] . jQuery also permits connect asynchronously with the server, through AJAX[12] calls.

- **Handlebars:** Handlebars[13] is a JavaScript library for building semantic and intuitive templates that facilitate the developers work to build their user interfaces. Handlebars is compatible with Mustache template system.
- **Bootstrap:** Bootstrap is the most widely used framework for front-end developing. It contains HTML and CSS-based design templates for typographic forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. The mostly components and features support cross-platform and support the most used browsers in the market.

3.3. Server

The server side of the application is responsible for storing all the data in a unique resource, moreover it provides services for the application interaction with that same data. Therefore, we need to design an architecture capable of support simultaneously data storage, provide a quick access to data and be of easily integration with every standard application, such as web and desktop.

We use a MySQL relational database to storage and retrieve all data and simultaneously keep the data consistent over the time. To ensure a secure and homogeneous access to all the application information we need to create methods and services that will handle and manage all the data. Thus, we develop the several methods in Java, creating RESTful[14] web-services. They are deployed and available through Netty[15] web server.

[10] http://www.similartech.com/categories/javascript
[11] http://w3techs.com/technologies/overview/javascript_library/all
[12] AJAX (Asynchronously JavaScript and XML)
[13] http://handlebarsjs.com
[14] REST – Representational state transfer
[15] http://netty.io

The use of REST web-services ensure an easier and faster access to storage data and also simplify the integration with several development platforms.

RESTful APIs[16] follow a set of standards [18]. They are defined using a base URI[17] (e.g., https://localhost:9443/patient/), an Internet media type (e.g., JSON[18], XML[19]) to send data between client and server side and also standard HTTP methods (e.g., GET, PUT, POST, DELETE). In the table 4 we summarize how these methods are used.

Table 4 - How HTTP methods implement a RESTful API

Resource URI	https://localhost:9443/ patient/**id**
GET	Obtain information of the patient according with the requested **id** and the information is presented in an appropriated Internet media type.
PUT	Update data of the patient according with the **id.**
POST	Insert a new patient in the database (do not use **id** in this method).
DELETE	Delete the patient with the **id**.

Thus, using a RESTful web-services we can provide several methods allowing an appropriated access to the application data, at the same that we grant a secure and controlled way for the client-side application to exchange data with the server. On the other hand, the application data and information are very sensitive. Thus, we need to carefully control the access to the web-service methods and hence, to the database. Therefore, and since we have users with different access permissions, we need to control the access to the different web-services methods. So, to ensure that this role is correctly applied, we incorporated a RBAC[20] plugin, developed in the University of Aveiro by the bioinformatics group[21]. This plugin filters the access to specific methods according to the user permissions and guarantee that any change in

[16] API – Application programming interface
[17] URI – Uniform resource identifier
[18] JSON – JavaScript Object Notation (http://json.org)
[19] XML – Extensible Markup Language (http://www.w3.org/XML)
[20] RBAC – Role based access control
[21] http://bioinformatics.ua.pt

the database is performed by a user with the right permission, preventing possible attacks.

In order to ensure a full protection in the data transference between client and server side, we use HTTPS[22] over HTTP methods because guarantee that the data transference occur in an encrypted channel.

3.3.1. Services

We develop Java web-services with the objective to ensure a correct and secure exchange of data with database. When we develop the Java web-services we had in attention to use Jersey[23] that is an implementation of JAX-RS[24]. Jersey simplifies the development and deployment of web services.

The several developed services are deployed and available using Netty server. During the development we separated services by different packages:

- **Account**: in this package we have methods that allow showing and managing user data, like update user data;
- **Login:** this package provides methods for controlling the access to the system;
- **Manage:** this package represents methods related to the system administration, where only administrator has access to these methods. So, the administrator can access methods like create organization, edit organization, add devices, edit devices, invite users to the organization, edit users ,etc.;
- **Patient:** the patient package offers methods that allow managing patient data. Thus, this package provides methods like create a consult, assign user to a consult, add a patient ,etc.;
- **Simulator:** this package provides methods that control all the communication between HemoSpec Platform and HemoSpec Device;

[22] HTTPS – HyperText Transfer Protocol Secure
[23] Jersey – https://jersey.java.net
[24] JAX-RS – Java API for RESTful Web Services

- **Details:** this package represents methods related to patient analysis. So, it represents methods like start an analysis, technician validate analysis results , technician write comments associated with a specific module results or physician submit analysis results;
- **Device:** this package provides all methods related to the device that represents the interaction between device and users, like stop device, reset device, clean device, etc.;

3.3.2. Data structure

In this sub-section, we will show how our database is structured and organized. The database is centralized in the organization (Figure 23), since all information in the platform is dependent of the organization.

An organization can have multiple associated roles, which are assigned to different users. Since, to each user is assigned roles according to their respective privileges. Thus, only users with specific roles may interact with the device organization. On the other hand, only users with the Administrator role can add users to an organization. In turn, each patient can have multiple consults associated, which informs users about the state of the patient. Their consults are simultaneously associated with the organization.

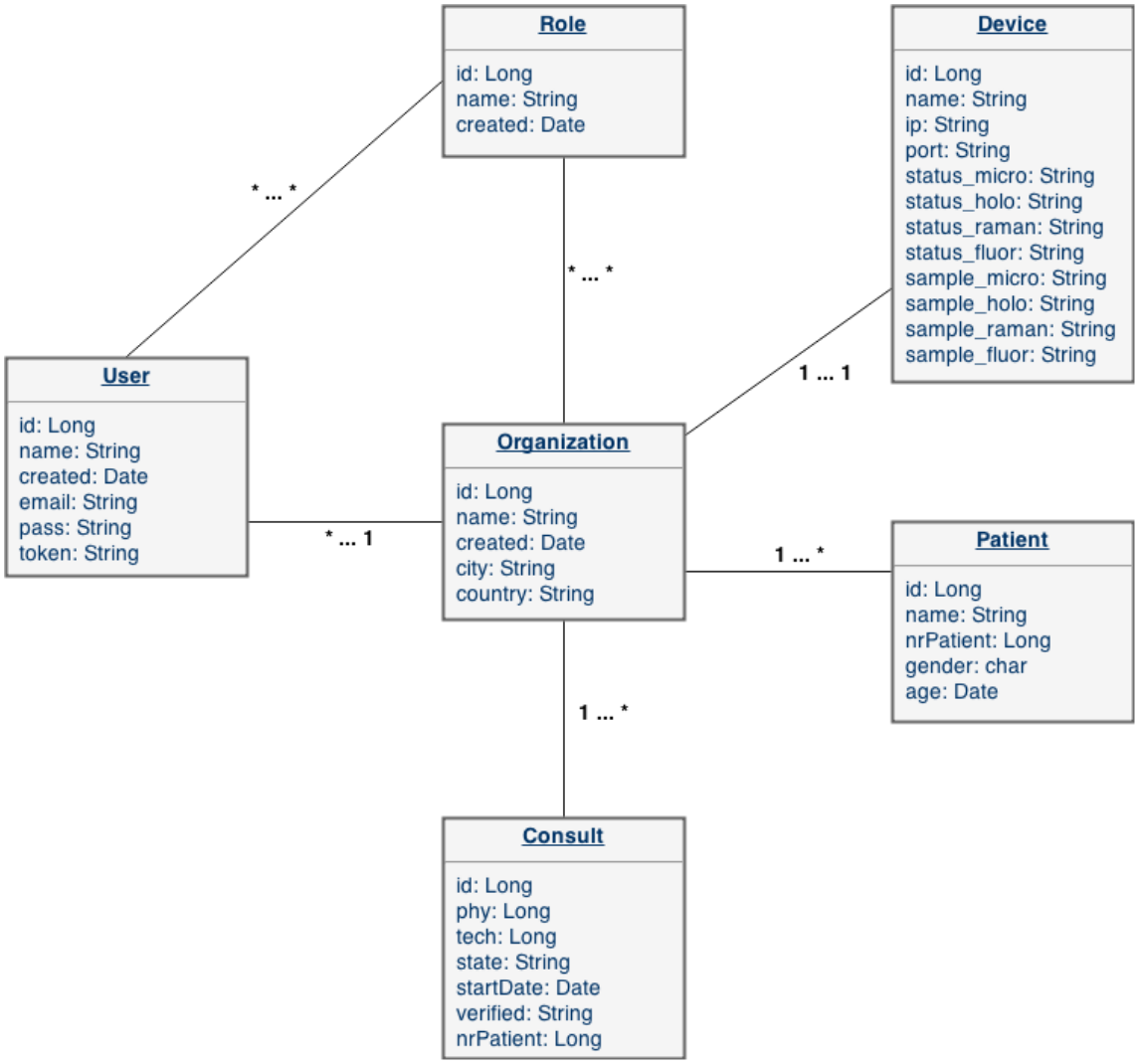

Figure 23 – Database relations with organization

As we referred in the last paragraph, one organization can have several roles. However we can see that each role is different from another, since each role has specific permissions associated. All permissions are different between them because each permission has associated one specific operation and one specific category (Figure 24).

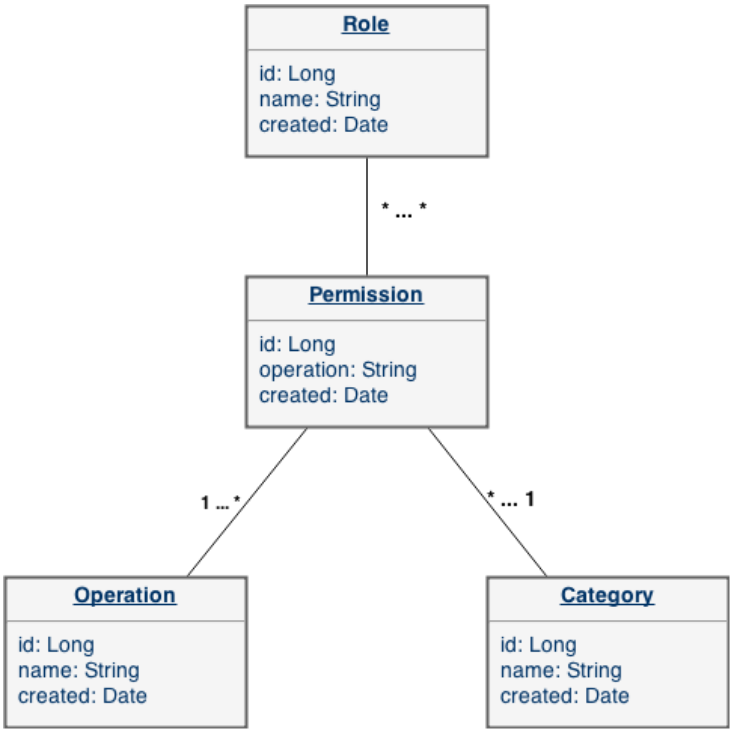

Figure 24 – Database sub-structure imported from RBAC plugin

Now, that we already described all direct relations of the organization and role, we will describe the relations associated with a consult (Figure 25).

A consult has associated one historic, in which are registered all the actions executed under the respective consult. On the other hand, a consult also has associated one analysis that contains all data provided during the analysis process.

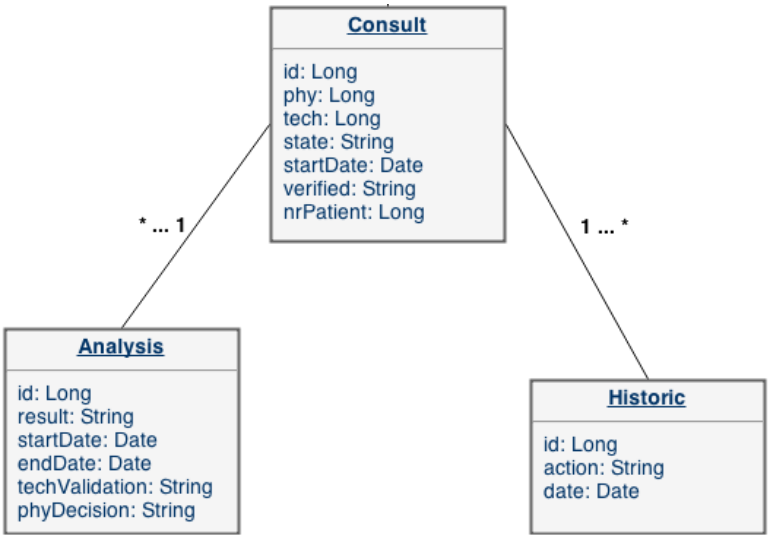

Figure 25 - Database relations with consult

As described in this document, one analysis is elaborated by four different modules, which implies that is necessary to store all the information associated with each module (Figure 26).

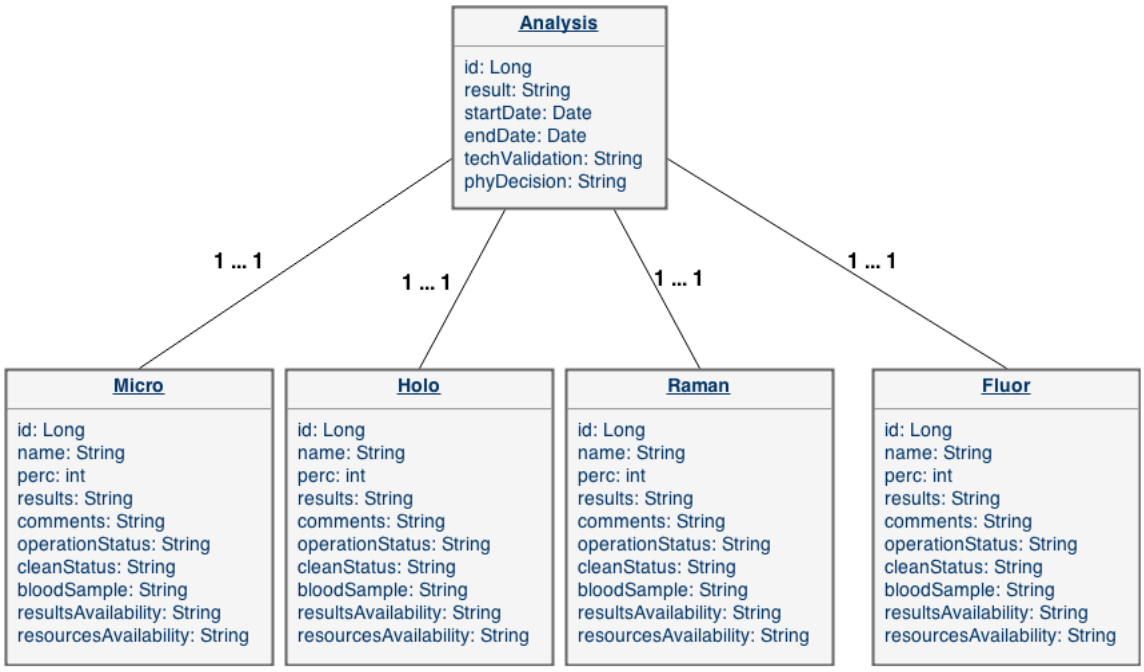

Figure 26 – Database relations with analysis

3.3.3. Communication protocol

The communication protocol guarantees that the exchange of data is done in a proper way, respecting all communication requirements. So all communications will be performed through REST web-services, on top of HTTPS in the way to ensure data security and privacy. The communication protocol is implemented over a master-slave architecture [19], where we have one master (software controller) and four slaves.

The communication protocol uses just one data format that enables a consistent communication throughout the platform. However, this does not affect the way like each slave produces its own data, since each slave will be associated with one communication interface (Figure 27), that is responsible for doing the translating from the data format produced by the slave to the data format applied in the communication protocol. Each communication interface is not more than a simple HTTP server that implements REST services.

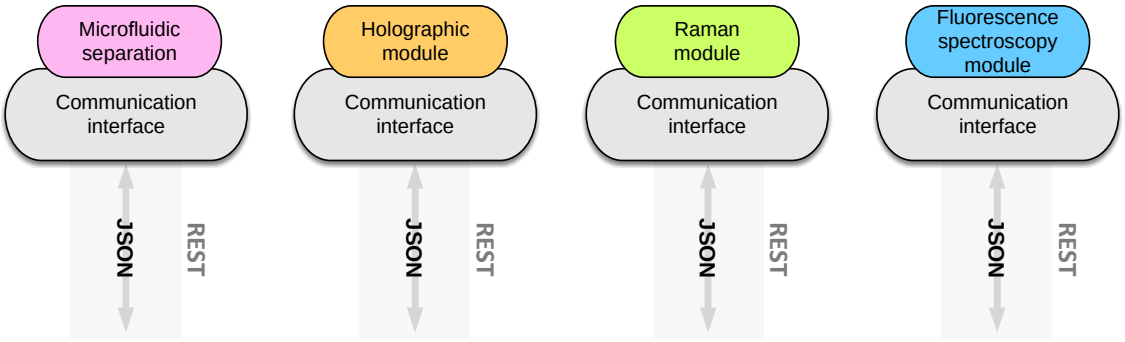

Figure 27 - Implementation of communication interface in each slave

In the Figure 28 the master, Software Controller, represents the server side of the HemoSpec Platform, while each slave represents a device module. We can see that every module can communicate with the software controller, enabling storage and data exchange. However, we also enabling the communication between slaves, avoiding that all communication passes by the software controller. For example, if the Microfluidic separation slave needs to send some data to the holographic slave, they just have to exchange data between each one. This fact is much important because communications

are considerably faster than using the software controller as proxy.

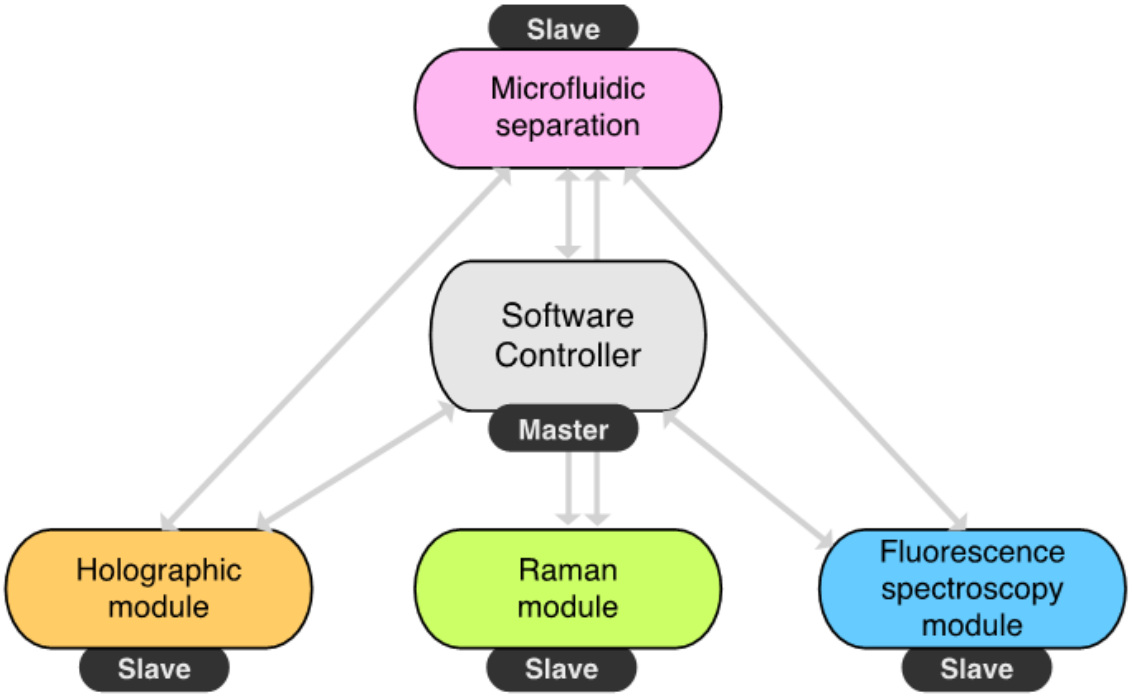

Figure 28 - Master/slave HemoSpec architecture

Once described the architecture master/slave used in our solution, we will now describe the various status that characterize each component. Also, we will describe which methods are used to obtain information associated with each components' statuses and which methods allow to provide information to each component, allowing to ensure a correct working of the solution.

Firstly, we will describe the different slave statuses, which express every action or slave's status. Follow we present a list with the various status:

- Operation status;
- Clean status;
- Blood sample availability;
- Results availability;
- Resources availability.

The "operation status" allows to control and verify in real time the operation associated with each specific slave. A slave has associated 9 different operation status:

- Ready – when the slave is prepared to start another analysis;

- Error – when occurs a problem during the analysis process;
- Lost connection – when the slave is disconnected from the master;
- Initializing – this status only occurs when the slave is turned on, when execute initial configurations;
- Processing – this status represents the execution of an analysis;
- Cleaning – a slave has associated this status during the clean process;
- Sending – when one slave sends information directly to other slave;
- Receiving – when a slave receives important information from other slave to continue the process;
- Waiting – a slave already finished the analysis process and wait for new orders;
- Stopped – when a user stop the slave;
- Finished – when the slave finishes the analysis process.

The second status presented above is the "clean status". This status informs if the slave is clean or dirty. After a slave starts an analysis the clean status changes to dirty. On the other hand, if a specific slave is cleaned the status change to clean.

The status described in the last paragraph is much affected by the "blood sample availability status". This can exchange between 2 different values, with or without sample. So, when a slave is clean, meaning that the respective slave does not have any sample associated, on the other hand, when a slave exchanges the clean status to dirty meaning that the slave has the sample associated.

During the analysis process can be necessary has additional resources associated with the slave. This way, each slave has associated the "resources availability status" that indicates if the slave has any resource in its possession (with or without).

Finally, we will describe the "results availability" status. Basically, this status indicates if the slave already has in its possession results or not (with or without).

After we had described the different statuses used to control all the process, we will present the various methods types that control all communications during the process, all the methods are executed by the master. In specific, we have 6 different types of methods.

The first method type is the “availability” method that allows informing the master about the availability of each slave (on, off, ping, etc.). The second method type is the “state” method that expresses the state of the device (initialized, cleaned, reseted, etc.). Another method type is the “status” method that allows to “get or set” statuses associated with different slaves. The device also has associated the “blood sample” method type that provides information about the analysis process. The fifth method type is the “data exchange” method that permits 2 slaves communicate directly without contacting the master during the communication. Finally, the last method type is the “configuration parameters”, which allows to get or set parameters of a slave.

Summarizing, with master/slave architecture we allow that each slave communicates simultaneously with the master (software controller) and between themselves enabling consistent communications.

Chapter 4

HemoSpec Platform

In this chapter we describe the HemoSpec Platform, a web-based platform that was developed to fulfill the requirements identified in the chapter "Requirements analysis". First, we will describe the main actors as well the workflow that was used to guide the system development. Afterwards, we will describe the communication protocol used between the platform and the device. After this, we show and describe the user interface and finally we explain how we implement security in our services.

4.1. System description

HemoSpec Platform is a web-based platform with the objective to create a solution that centralizes and speeds up the treatment process of sepsis disease. In the way, to speed up the process, all analysis will be done in a unique device that exchanges directly results with the platform. The data exchange uses a communication protocol that will be described later. Do not forget, that the developed platform was designed with focus in the usability and user friendliness, in order to ensure that the interaction with users will be simple. Also, we had in account during the development the necessity of implement optimized methods to guarantee a fast and efficient solution processing.

In our solution, each user, device or patient is associated with one specific organization (Figure 29). In one organization, we can have three different types of users, from the administrator until the technician. The physician and the technician have the responsibility of controlling the consult and device

workflow which are associated with a patient's study. While the administrator besides of these responsibility already described, he still has different responsibilities associated. He is responsible for managing the organization and for managing users associated to the respective organization. The administrator also is responsible for managing the device, for controlling patients associated with the organization and for defining roles associated with the organization (physician, technician, etc.). To finalize, associated with a consult we have one analysis and the respective historic that contains all the actions executed under the consult.

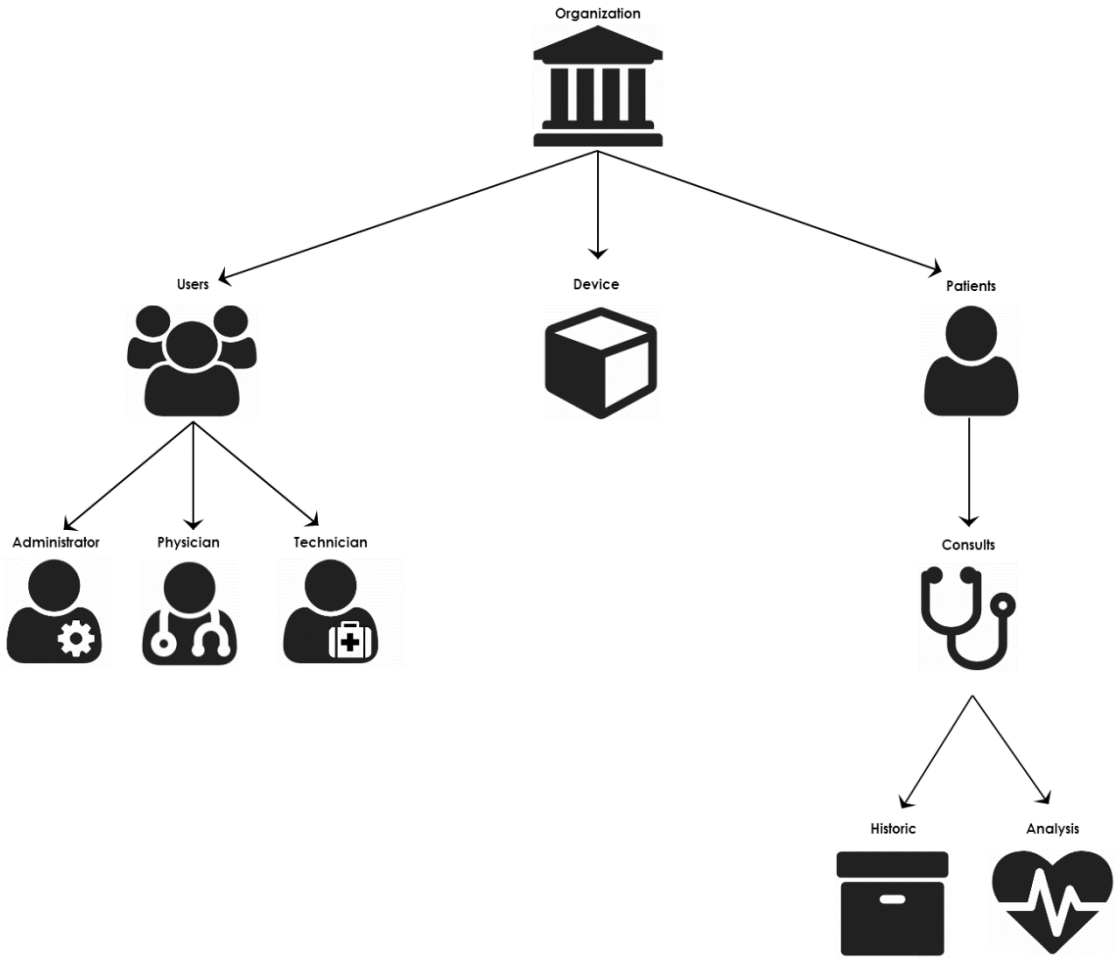

Figure 29 – Basic scheme followed by HemoSpec Platform

In the next three figures (Figure 30, 31, 32), we present the normal workflow followed by our application. So, we will start by the first figure that represents the initial phase of the workflow.

To start the normal workflow users adds a patient to the system, which respectively will be associated to one organization. After the respective user adds the patient, any user can continue the workflow. To continue the workflow a user only needs to click in the button "Request study" and automatically the patient study/consult will be initiated. At this moment, the patient consult status exchanges to Requested due to the click did before. The last step represented in this figure is associated with the click in the "Assign" button. This click will update the workflow, at the same time exchanges the status from Requested to Assigned.

Note, when we want to start a patient workflow and the patient is not in the database, like we assume in the last paragraph we have to add the patient. However, if the patient already exists in the system, to start the workflow a user just needs to click directly in the "Request study" button.

Figure 30 – First section of HemoSpec workflow

Now that a consult already is in the Assigned status, to continue the workflow, the technician has to start the analysis (Figure 31), by clicking in the "Start study analysis" button. When the technician clicks on the button instantaneously the platform turns on the device that processes the analysis.

During the analysis process both users can visualize the consult and device status. However, only the technician can manage the device during the analysis process. When the analysis terminates, the system change automatically the consult status to waiting technician. After, the technician going to take a decision according with the observed results. Depending on

results, the technician can do a reanalysis what going to repeat all the process or if results are good can validate them. Since technician validates results, the platform will assume this decision and going to change the consult status to waiting physician.

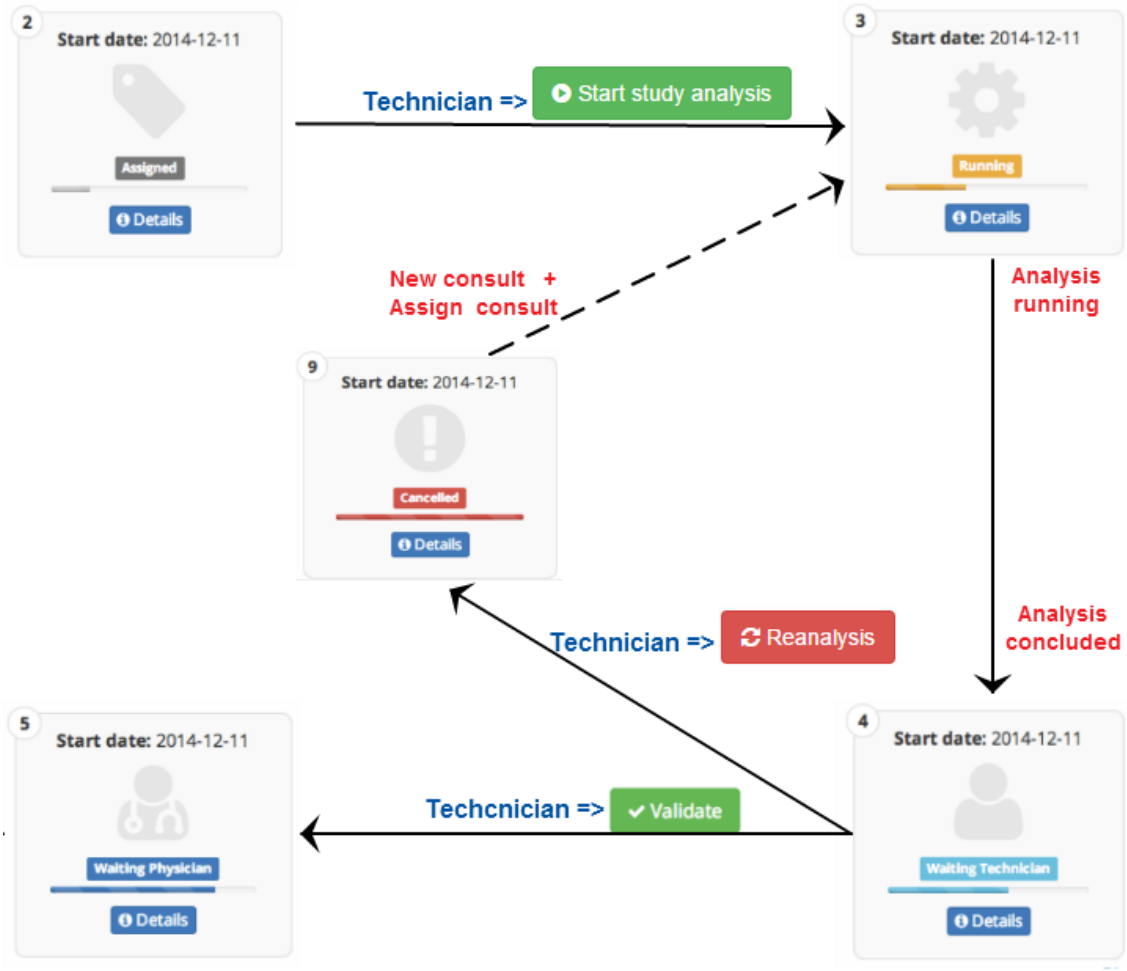

Figure 31 – Second section of HemoSpec workflow

The last step of the workflow is based on the physician decision (Figure 32). First the physician will see the analysis results and the comments of the technician. Finally, the physician will take a decision according with results.

Figure 32 - Third section of HemoSpec workflow

4.2. Communication protocol description

A communication protocol is fundamental to assure a successful transmission of data in a system, where is respected the data format, the speed of transmission, the quantity of data exchanged, etc. All these restrictions guarantying a consistent communication in a system.

In our solution to ensure consistent communications and that all rules are respected, the communications between components are always started by the master. Thus, the master has control and knowledge about all communications that are occurring in the system. And every time, the master can intervene in the communication, guarantying that all is fulfilled correctly.

As we described in the section 3.3.3, in our platform the modules can execute several different methods to satisfy all necessities. But, it is not objective of this document describes how all these methods communicate. So to avoid the descriptions of all these methods, we group the various methods by different groups. More in concrete, we have 4 groups that symbolize all the different communications that occurs in the system. Follow we will describe one method associated with each group and simultaneously presents a basic diagram that shows the communications executed by the method.

The first method (Figure 33) that we will describe is the “ping” method. As is any communication, the master initializes the communication with the slave. The master invokes the “ping” method, subsequently the slave receives this invocation and will process the respective method. When the slave finishes the processing, he sends the response to the master. After the master

receives the response, he updates the information and continues the respective workflow.

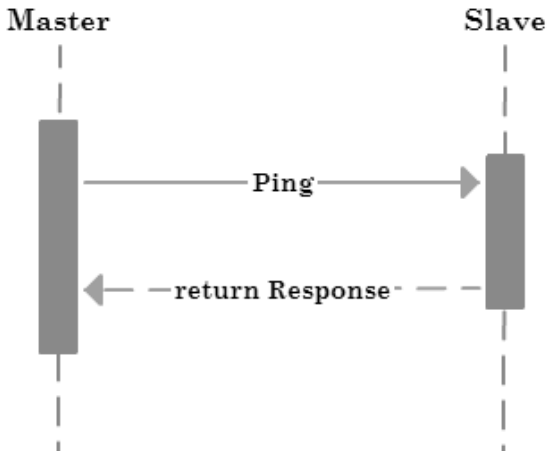

Figure 33 - Description of the ping action using the communication protocol

The next figure describes the "parameters" method, which has the objective to update the slave data. This way, the master sends information to the slave invoking the respective method. Follow, the slave will update its data with the parameters received from the master. Finally, the slave will advise the master that already updates its data.

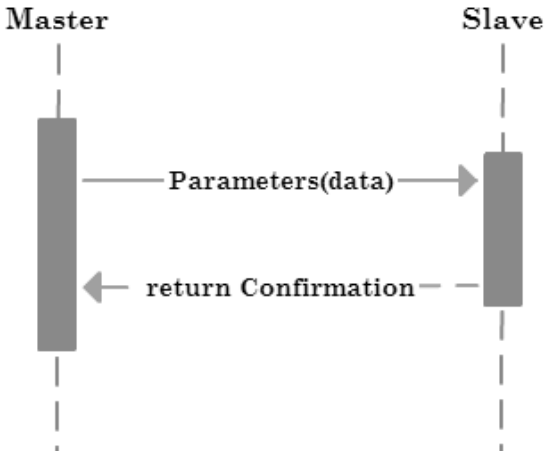

Figure 34 – Description of the set parameters action using the communication protocol

In the Figure 35, we will describe the clean action executed over a slave. More one time, the master starts the communication with the slave, this time through the invocation of the clean method. When the slave processes this

method, he will change the operation status to "Cleaning" and following return the operation status to the master. Afterwards, the master will invoke the "status" method to know if the slave already terminates the respective operation. So, when the slave terminates the clean operation, he will change several status associated to him like shows below. This way, when the master invokes this method and the slave finishes the operation, he will return the several statuses changed, updating the master data.

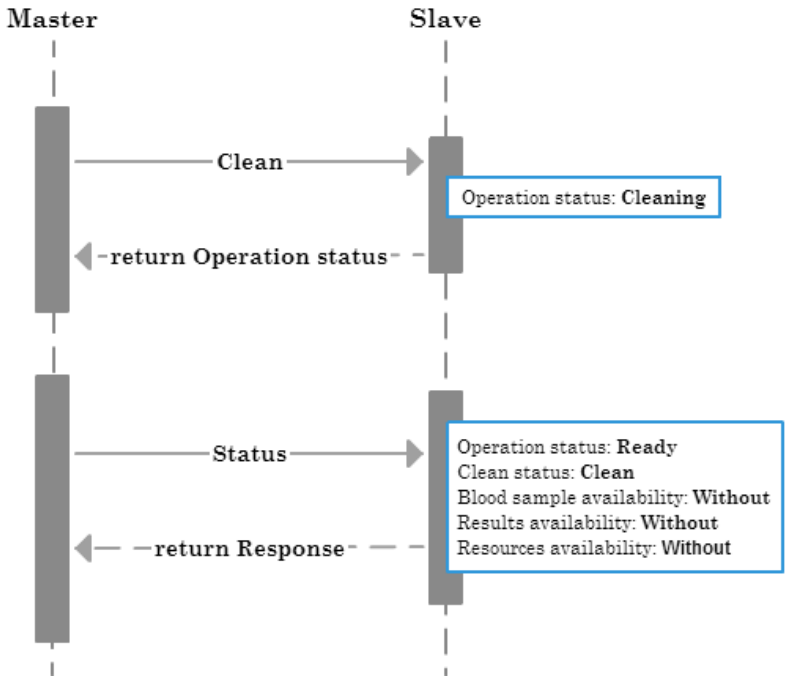

Figure 35 - Description of the clean action using the communication protocol

The last method that we will describe (Figure 36) is related to the data exchange between slaves. This method has the advantage of not involving directly the master, since this only tells initially to the slave 1 to send data to the slave 2, what allows that the communication will be faster than compared if was needed to use the master as proxy. Afterwards, we will describe all the communications realized to execute this action.

Firstly, the master invokes the "send" method to the slave 1 where informs who is the slave that will receive the data. After the slave 1 receives this invocation, he will change the operation status to sending and he will establish a communication with the slave 2 through the "receive" method. When the slave 2 processes this message, he will change the operation status

to receiving and send the confirmation message to the slave 1. After the slave 1 receives this confirmation, he also will send a confirmation to the master. While the exchange of data process occurs, the master continues invoking the "status" method that we already previously mentioned, but in this case there is no data to update (get).

After the exchange data process finishes, the slave 2 exchanges the operation status to waiting and invokes the "status" method to the slave 1 (set), where send the respective operation status, allowing that slave 1 update also its operation status.

From this moment, when the master consults the status of each slave, they send the new operation status and respectively the master updates data.

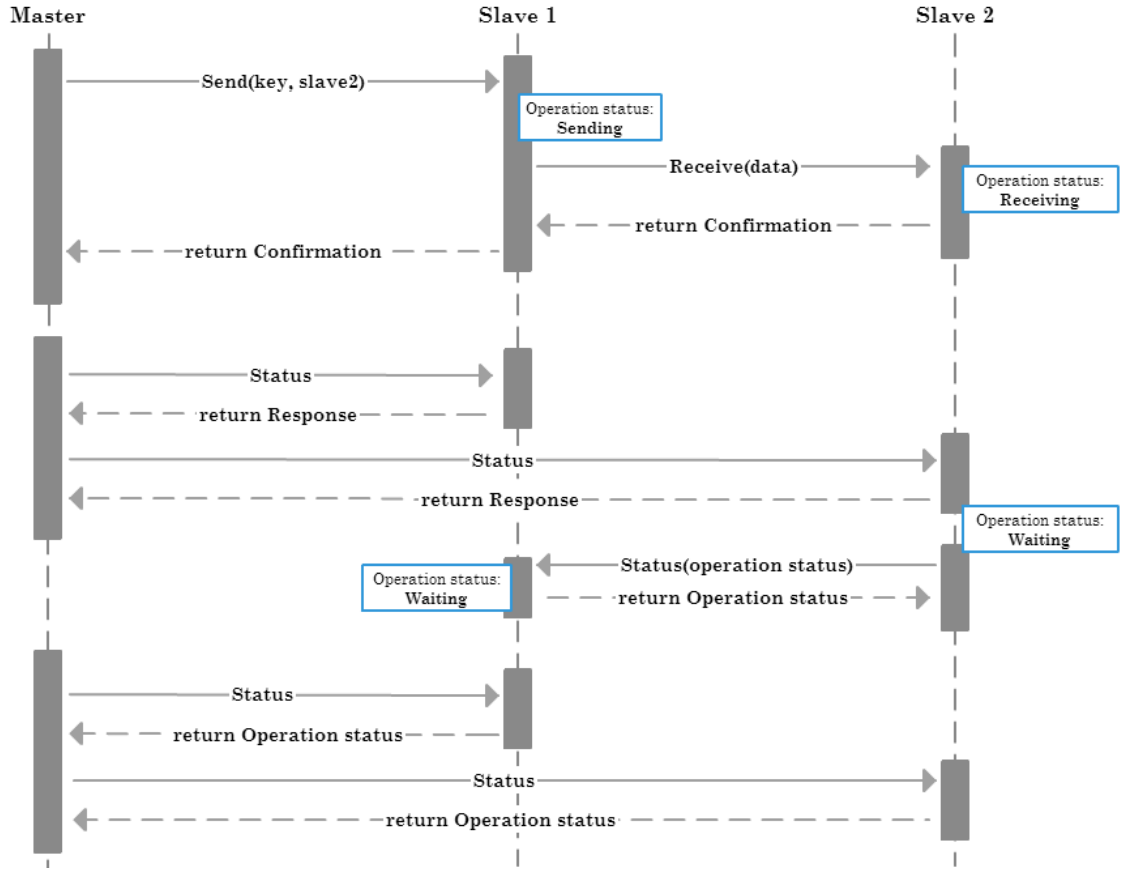

Figure 36 - Description of exchange data between slaves using the communication protocol

4.3. User interface

The user interface is much important in our solution, because it is through of the interaction with the platform that users interact with the system. If the interaction between users and the interface is not good enough, users do not take advantage of the solution. This way, it is necessary to have into account the usability of the platform and at the same time ensures that the platform is user-friendly, guarantying that interactions will be intuitive and direct.

In this section we will present details of HemoSpec Platform's user interface and respective interactions. Also, it is important to refer that we will describe more pages than we showed in the mockups section. We created more these pages with the objective of complements our solution.

4.3.1. Patient page

After we make login (Figure 37), the platform redirects us instantaneously to the patient page.

Figure 37 - Login page

The patient page (Figure 38) is the base page of our platform, since this presents the basic and relevant information of patients. Thus, in this page we can visualize the information of each patient, in the current moment.

To facilitate the search for a specific patient or search for patients that are in a specific consult status like Requested is adequate to use the search tool, that apply a filter with the required parameters about patients data. Also, it is possible any user adds a new patient, provided that, the patient is not already added in the respective organization.

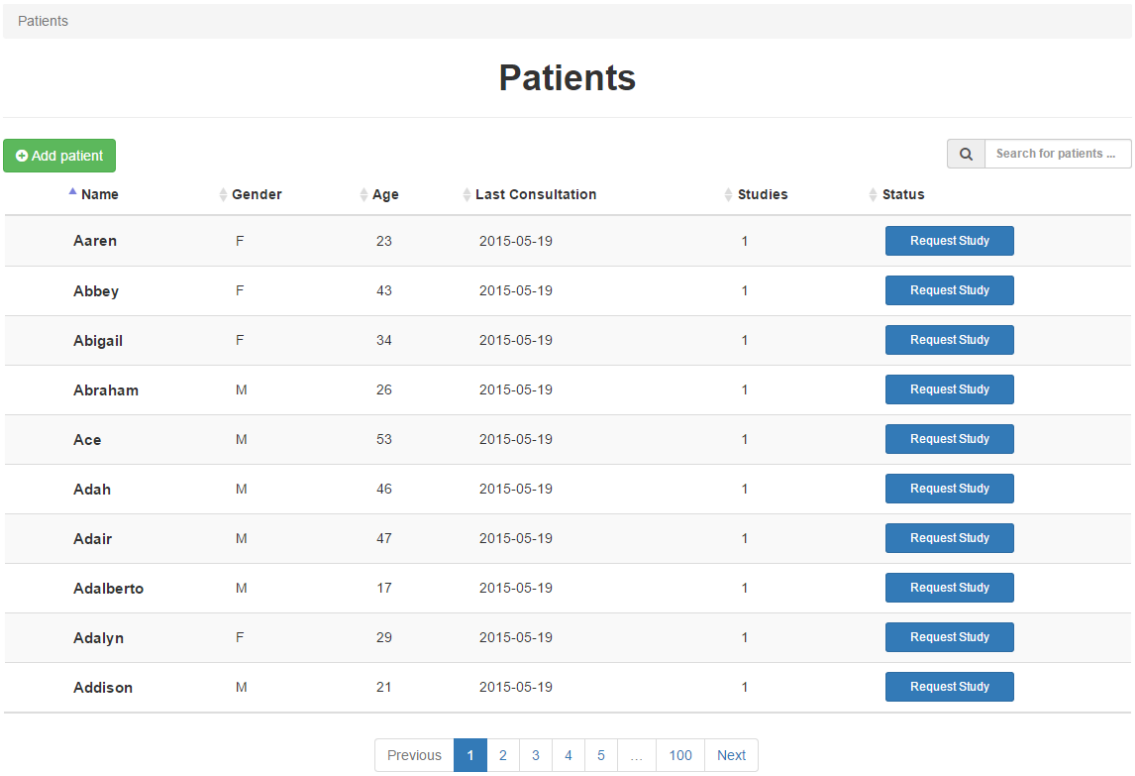

Figure 38 - Patient Page

As we can verify directly in the figure above, in specific in the first page of the pagination, does not exist any study in processing. This information is perceptible since all patients wait for a new study ("Request Study"). On the other hand, in the same page if there existed a patient in specific in the Requested status, this same information was visually acquired of a direct form because we saw a different label in the status column.

However, like we can understand in the system exist much patients associated. So we have more than one page with patients, what do not make clear of visualize if exist some patient's consult in the Requested status. Which implies that each user when do login in the system, he have to verify always if exist one or more patient's consults in the same situation.

To counter the situation described in the last paragraph, it was implemented in our platform a gender of a notification (Figure 39) that informs directly users, in specific technicians, that there are studies to follow (consults in the Requested status).

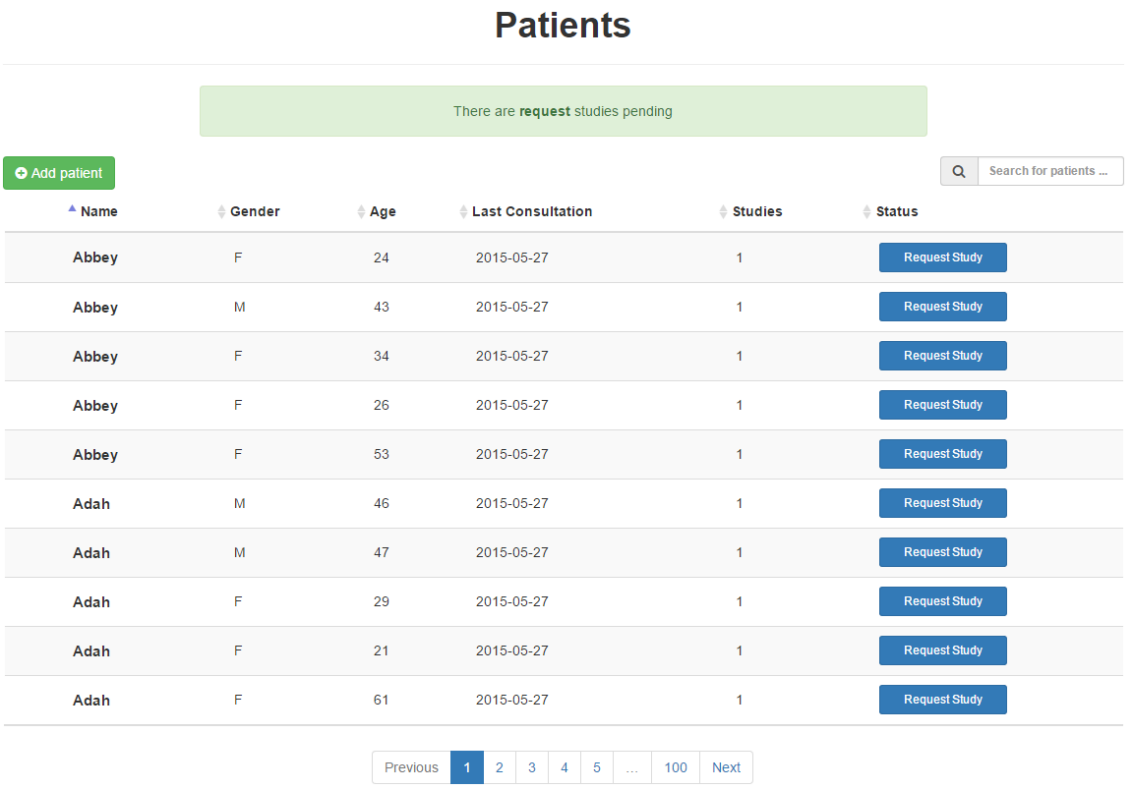

Figure 39 - Notification that there are studies pending

Assuming that there is one study in the Requested status (Figure 39) and this will be continued, then a user will click on the assign button. After he has clicked on the button, the status of patient consult will change to Assigned and at the same time, the notification presented in the last figure will be removed (Figure 40), since that this study is now being accompanied.

Figure 40 – Patient's consult is in the assign status

From this moment, the technician can start the analysis process associated with the study. To access the details page, where it is possible to start the analysis, the technician has to click on the details button associated with the respective consult (Figure 40).

4.3.2. Details page

Now, that we already described the patient page and following the HemoSpec Platform workflow (Figure 32, 33, 34), we will start for describing the details page.

After the patient's consult exchanges to assigned status, all users can access to the details page, but only technicians can start the respective analysis. Thus, when a physician access to this page and the consult is in the referred status, the platform disables the button "Start study analysis" (Figure 41) because the user does not have the right permissions to start the correspondent analysis.

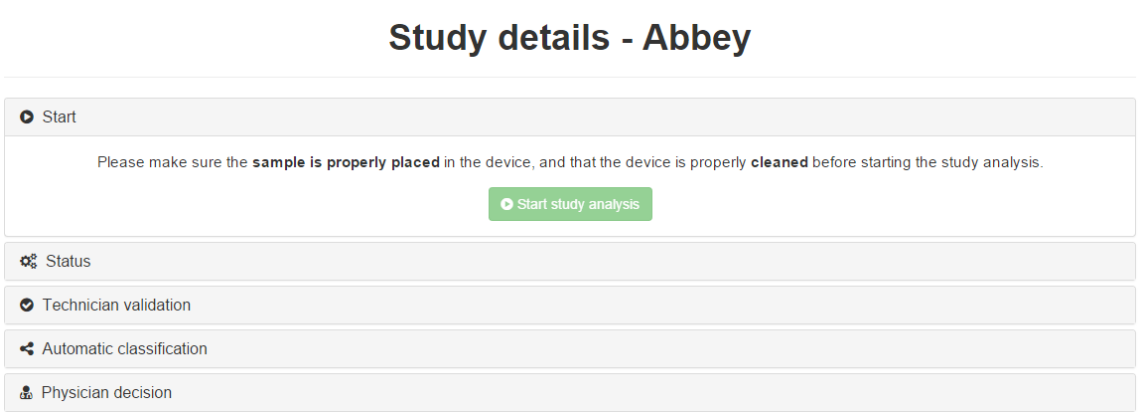

Figure 41 - Physician tries to start an analysis

Assuming that the technician accesses to the page to start the analysis, the platform renders the web page and at the same time it will see if some device is associated with the organization. If any device is associated, the platform presents an alert message (Figure 42).

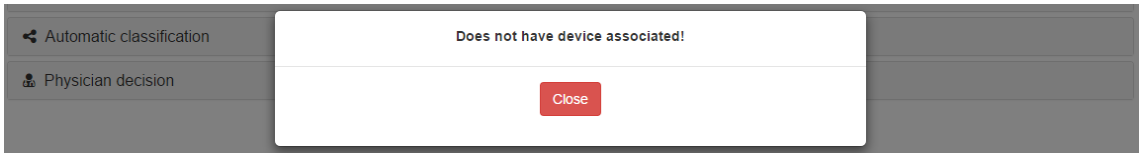

Figure 42 – Organization does not have device associated

After the platform finds that one device is associated with the organization, this will verify if the device is ready or not. If the device is not ready can show one of figures 43 or 44.

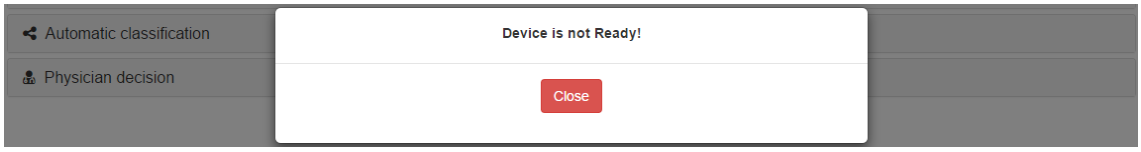

Figure 43 - Device is not ready to start an analysis

Figure 44 - Device does not contain the blood sample

After the platform makes all verifications and concludes that it is all ok to start the analysis, it presents the Figure 45.

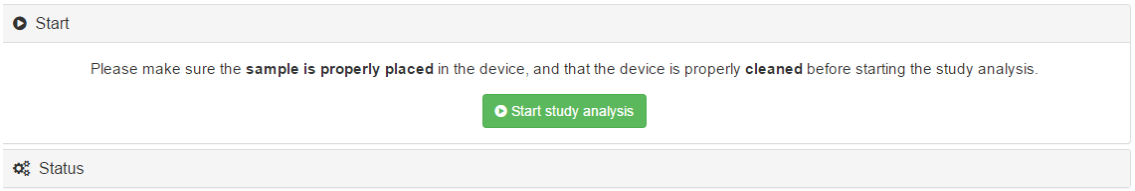

Figure 45 - Technician can start the analysis

Considering that the technician starts the analysis, instantaneously the platform realizes the respective actions. This way, the platform will be updated (Figure 46).

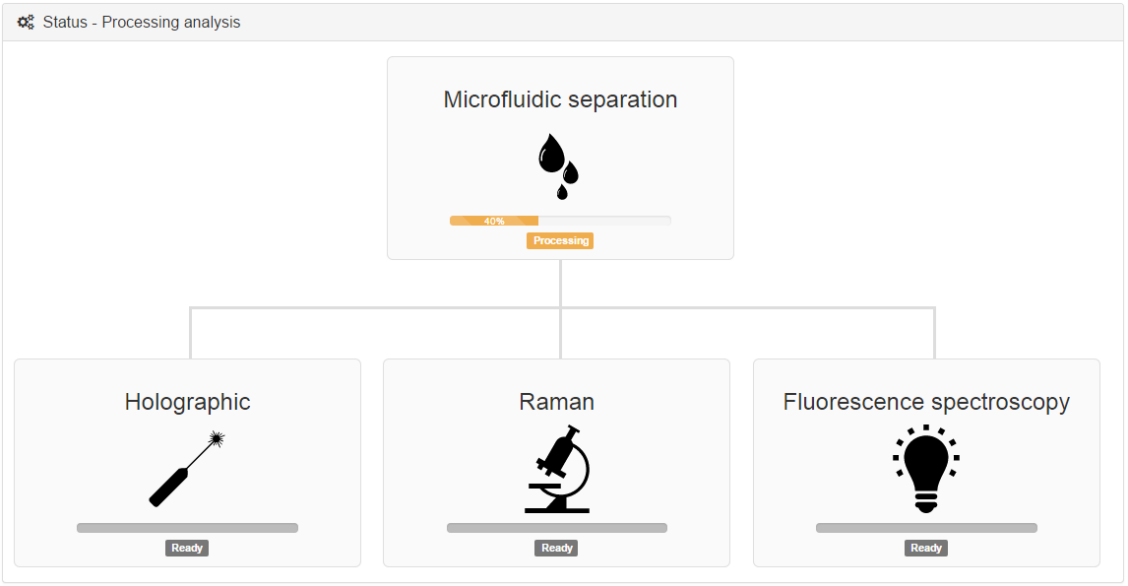

Figure 46 - Details page informs that the analysis is processing

During the update, the patient consults exchange the status, but now to running status and the device is started, like we see in the Figure 46. The device is responsible for doing the analysis and provides information to the platform. As we can see above, the analysis is divided in two parts. First, the device will make a separation of the different microfluidics presented in the patient blood. Just after, of this has concluded the separation is that starts the second part of the analysis. When the device terminates the microfluidics separation, it will update modules status. This way, the microfluidic separation module will finish the processing and will start the processing of the others (Figure 47).

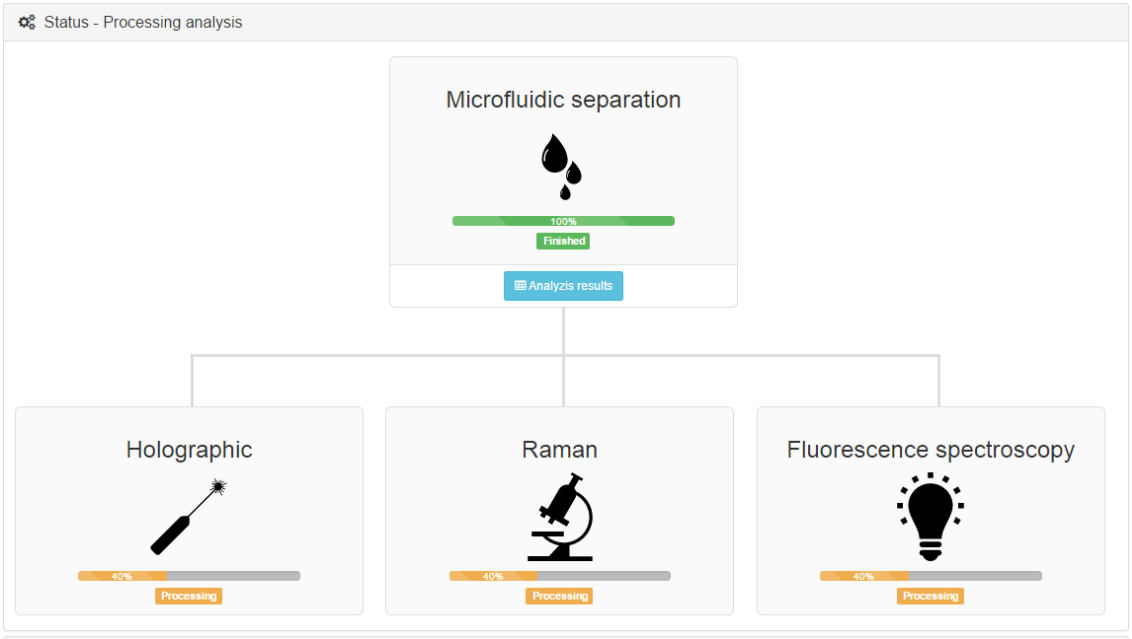

Figure 47 - Microfluidic separation module terminates its process and starts the other modules

Now, that one module has already terminated its processing, each user can consult the specific module results by clicking on the “Analysis results” button. After click on the button, the platform shows other web page that contains all the analyzed module information and where it is possible writes comments associated with the results. However, only the technician associated with the study is that can write the respective comment (Figure 48).

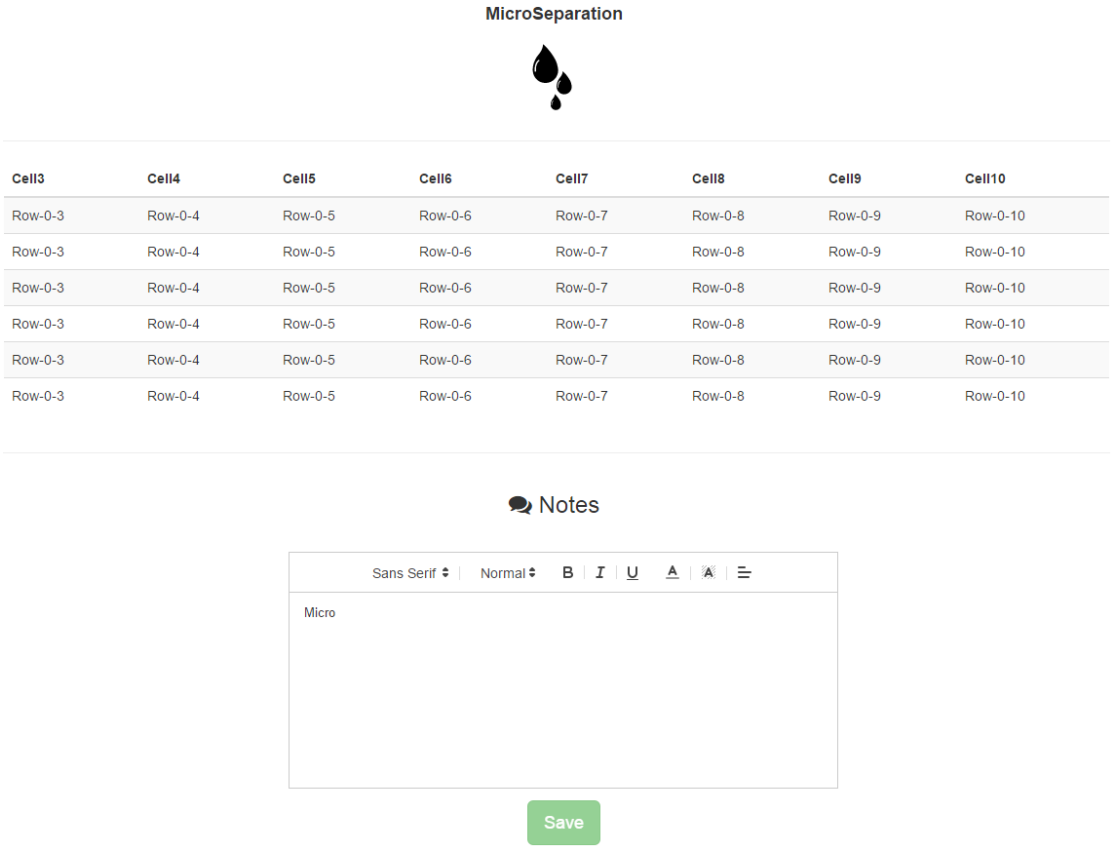

Figure 48 - Module details page, which presents module results and where the technician can write comments related to results

When the platform visualizes that all modules already finished the analysis instantaneously will update the data. Following, the platform automatically will present a new section (Figure 49), where the technician can write his decision according with the results and respectively select the correct choice for this study.

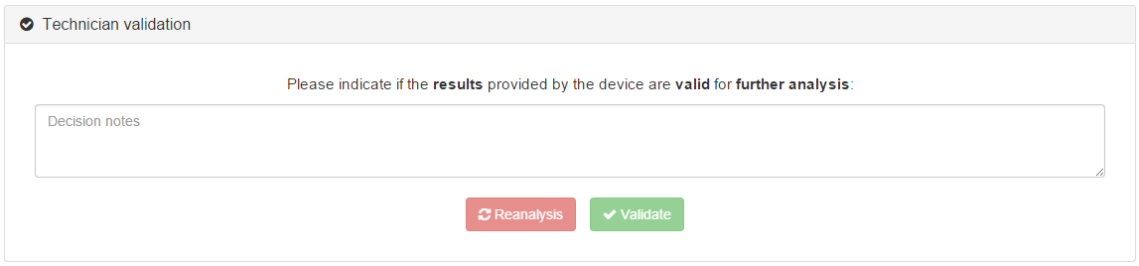

Figure 49 - Technician validation

Considering that the technician validates results, the platform instantaneously will assume this decision. At the same time will initiate an automatic classification of the results provided by the device. When this classification is concluded, the platform will present the results obtained to the physician (Figure 50).

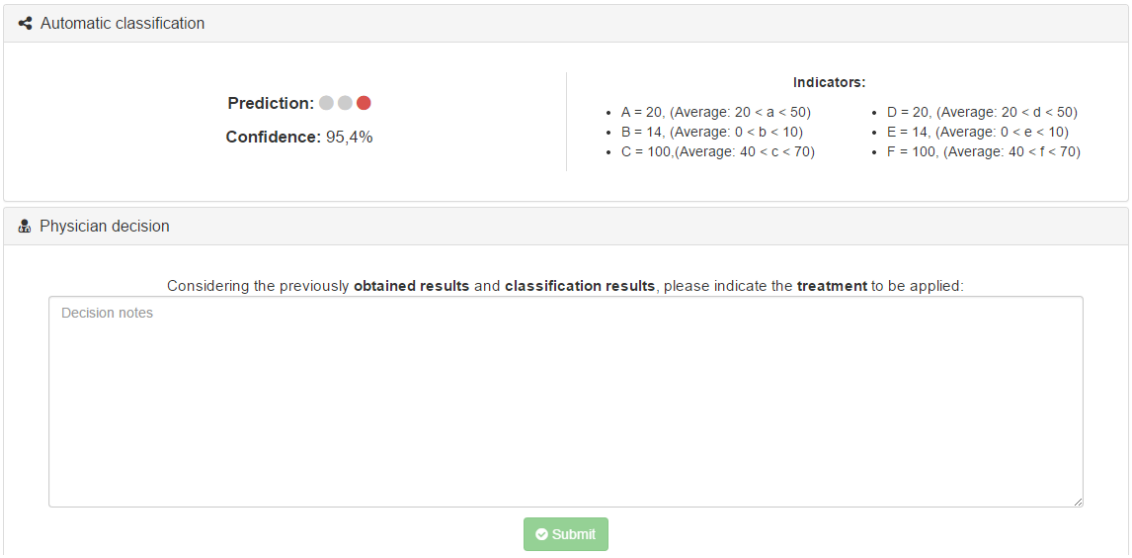

Figure 50 - Physician decision

Now that the platform presents the results classification of the current study, the physician already can take a decision according with the results to apply in the patient treatment.

4.3.3. Device page

On this page, the platform presents information about the device status, in specific shows the status of each module. Even though there is not any analysis processing we can see the information of each module. The several modules can be in different status.

This page is visible for every user registered in the platform, but only the technician and the administrator can execute actions over the respective device. As we referred each device module can present different status (Figure 51), what shows that each analysis done in each module is independent of each other. However, in this web page all actions taken by the technician or

by the administrator are executed over the device in general and not over each module in particular.

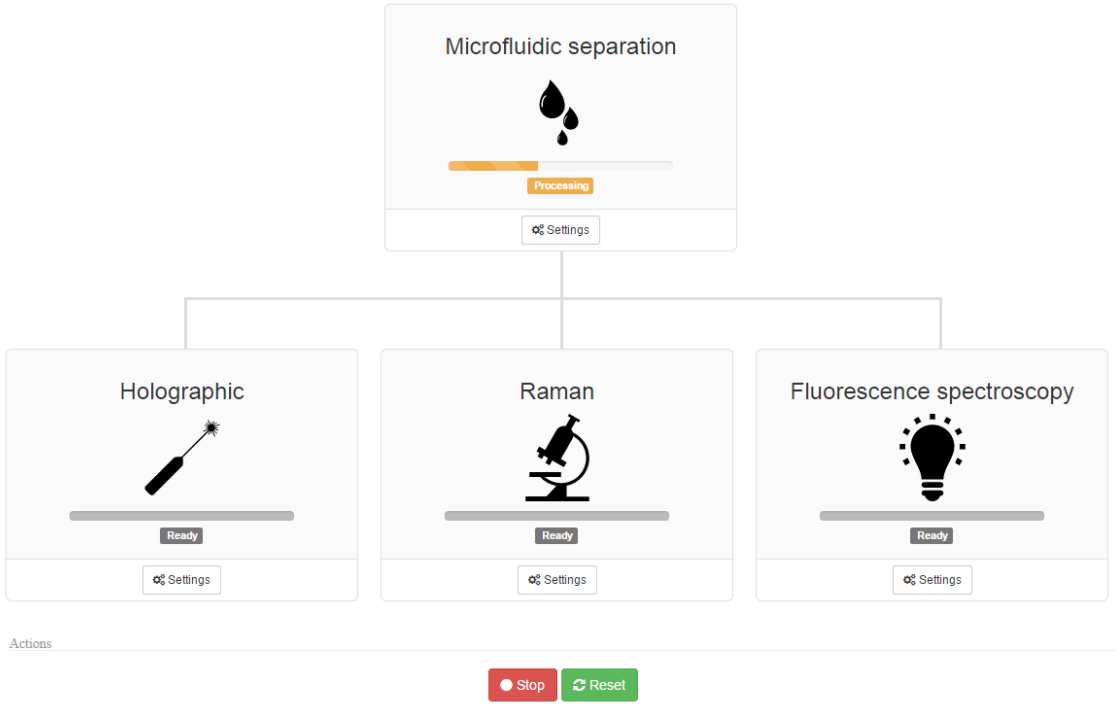

Figure 51 – Information of each module captured from the device page

About the device, users can execute different actions, specifically we have 3 different actions. The first is the stop action, represented by the Stop button, that allows breaking all the device processing, that is represented in the Figure 52.

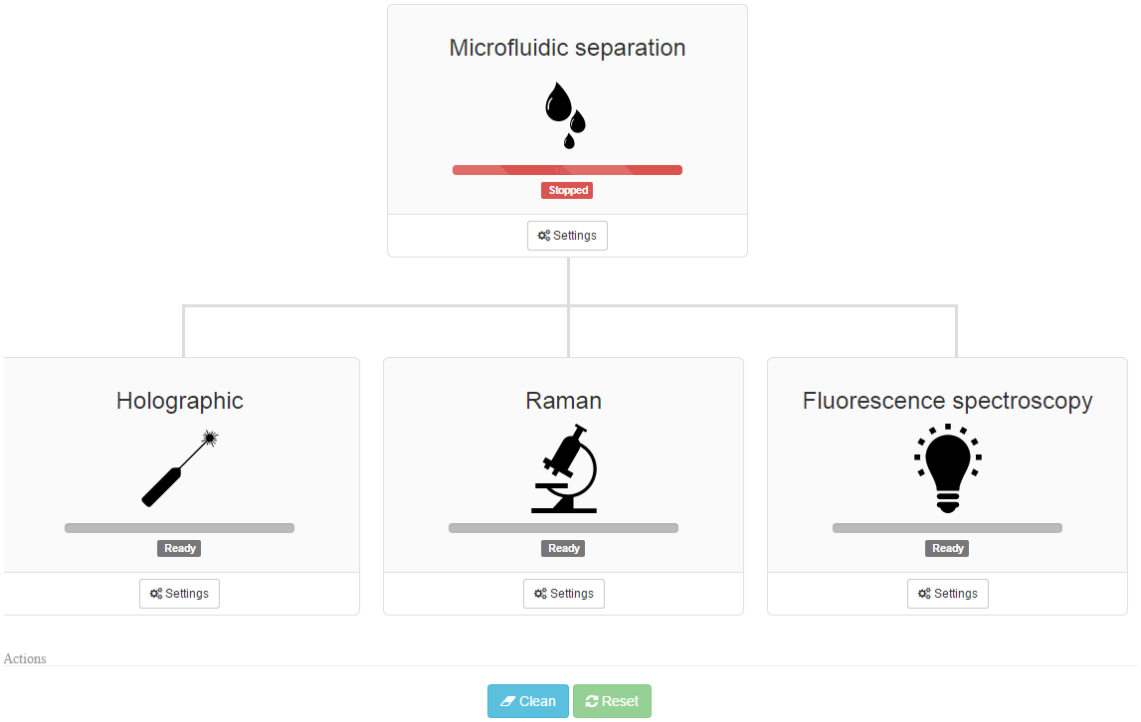

Figure 52 – Device status after executed the stop action

The second is the clean action that permits after finished the analysis or when the analysis is stopped clean the device (Figure 53). The last action is the reset, it allows that a technician can reset the device and simultaneously the analysis process (Figure 54).

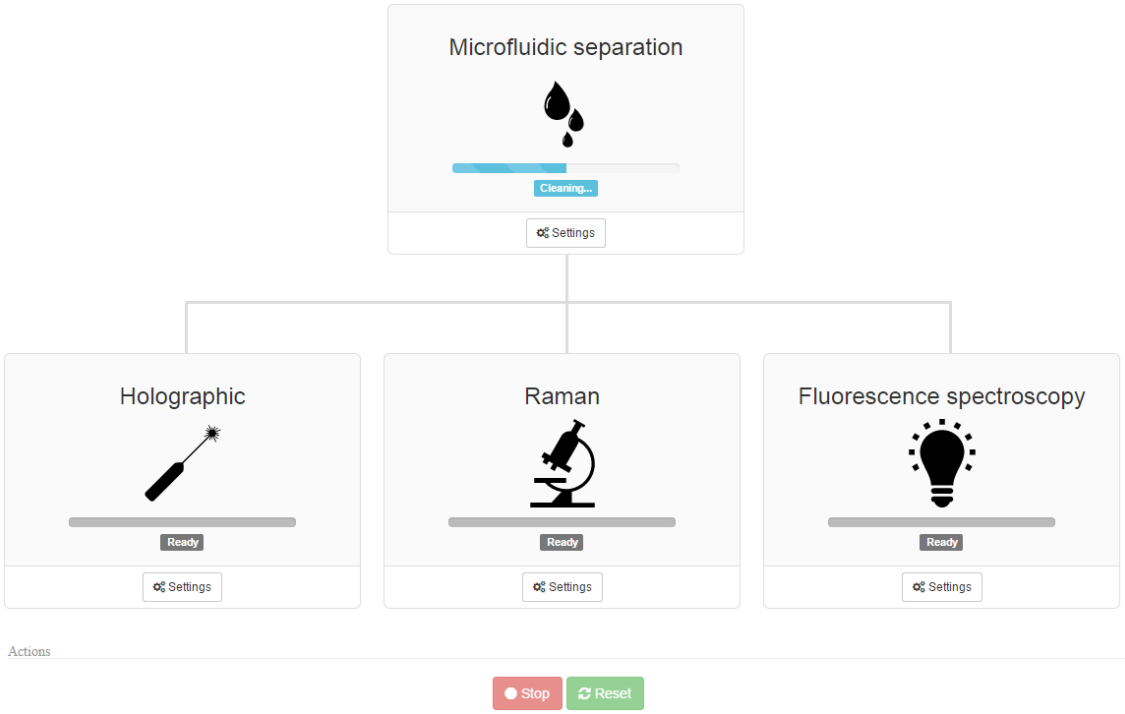

Figure 53 - Device status after executed the clean action

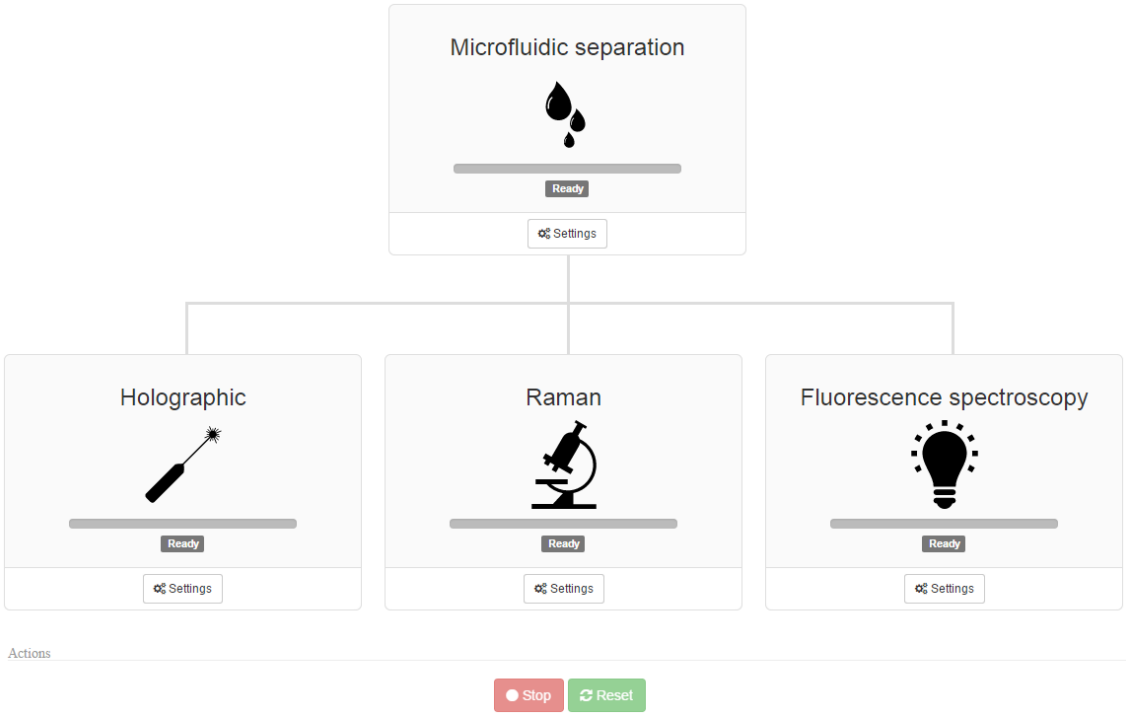

Figure 54 - Device status after executed the reset action

On this page, also it is possible to see the set of actions executed over the device. In each table row (Figure 55) is presented an action, the date of execution, the technician and the patient involved. The advantage of this list is that shows information of each action executed over the device. This way, when we need to know who is responsible for an action or when the action was executed, easily we consult this information. To facilitate the search method, associated with the table we have a search tool that allows to filter for any field associated with each row in the table.

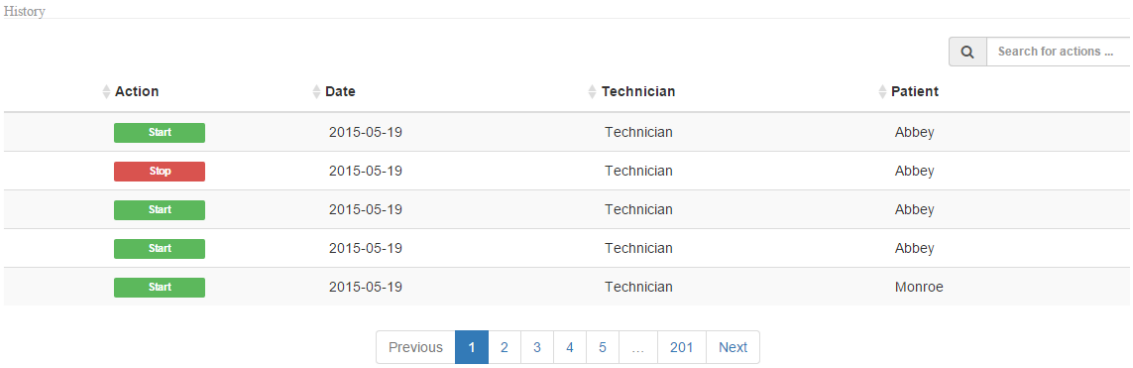

Figure 55 - list of actions associated with device

As we expressed, in the device page users always execute actions over the device in general. Whereby, if a user wants to execute an action just over a specific module, he needs to access to the specific module device page by clicking in the settings button associated with the respective module. In this page (Figure 56), users see all the information associated with the specific module device. The big difference between this web page and the device page is that in this page all the actions are executed only in this module.

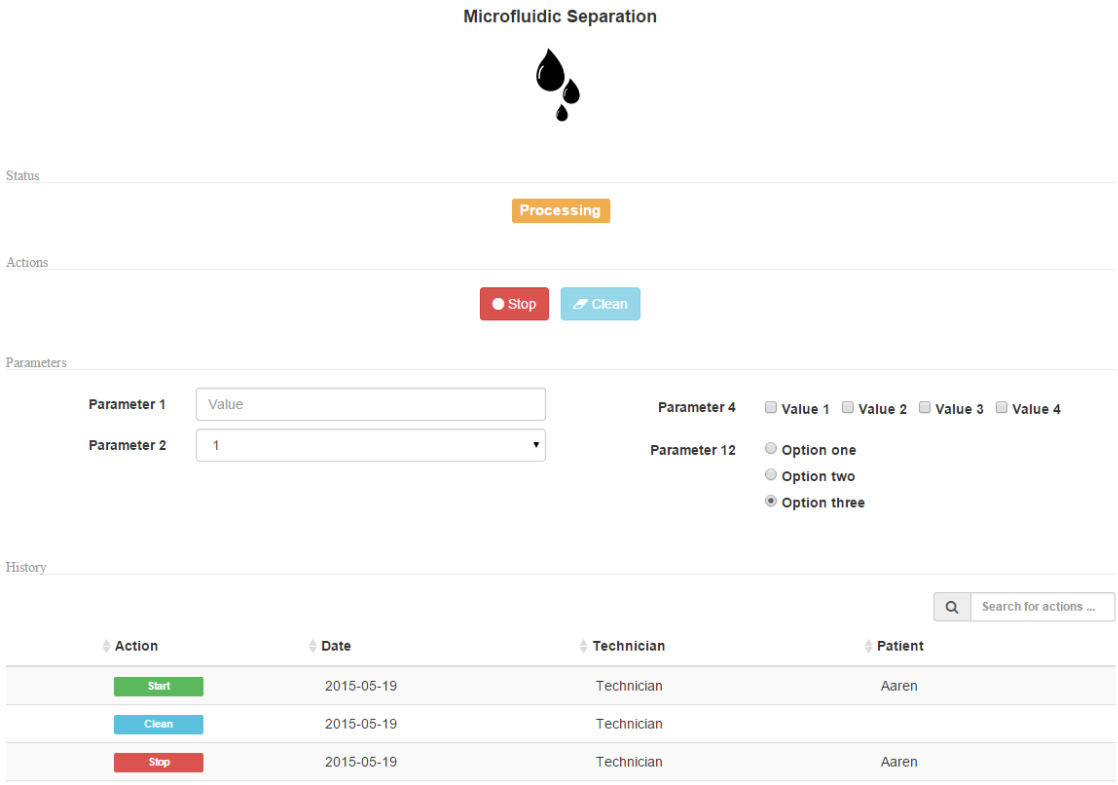

Figure 56 – Details page associated with a specific module

4.3.4. User account page

In the user account window (Figure 57), users can visualize own data and if necessary they can change their data. However, when a user changes his data, the platform verifies in real-time if the input data complies with all requirements. For example, when a user changes the email, it is required that the same follows a correct email structure or when a user changes the password is verified if the new password is different of the old password. The platform also verifies if the new password is a strong enough password to ensure the security on the platform. Only when all entered data is correct is that data can be submitted.

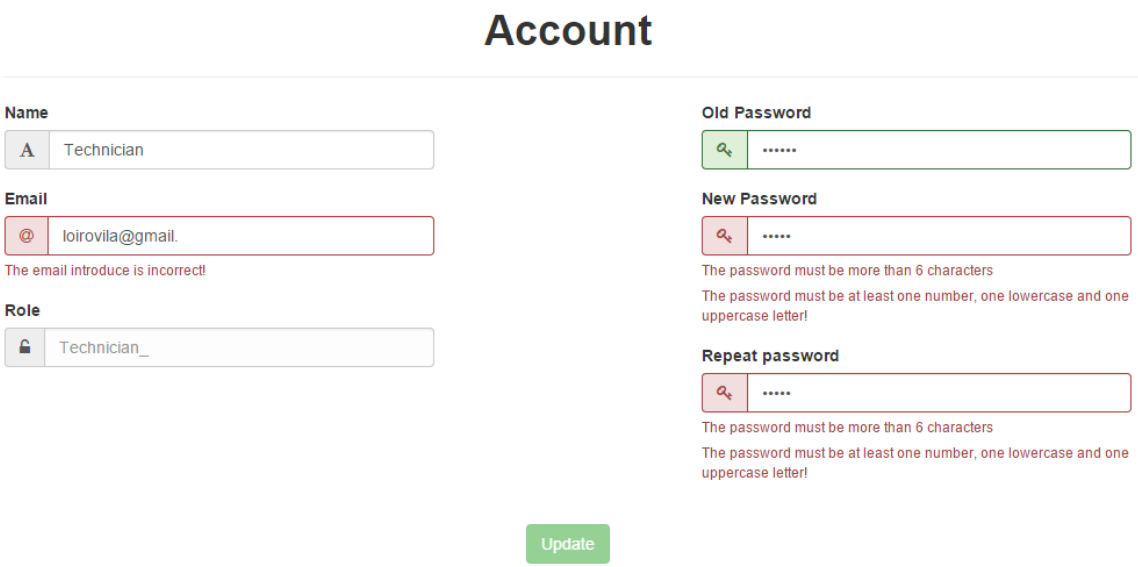

Figure 57 - User account page

4.3.5. Management page

Just users that have the administrator role can access to this page, because this is a restricted web page with the objective of control the organizations characteristics. As we can see in the next figure, just users with the respective role can see the Manage reference (☰) on the right side of the header.

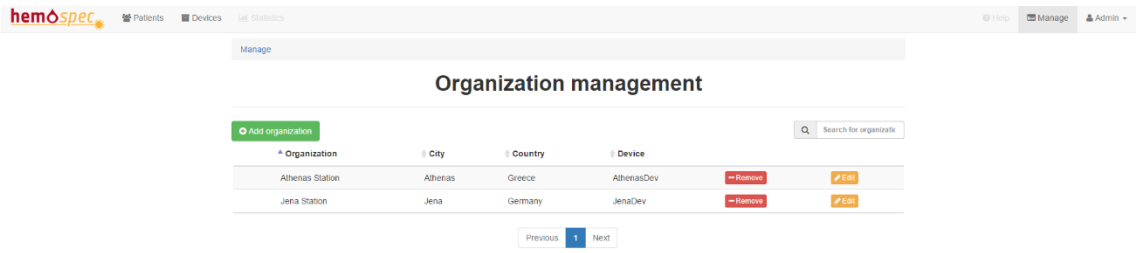

Figure 58 - Management page

When an administrator accesses to the management page, what this user sees directly on the page is the list of organizations associated with the system. After the administrator analyzes the presented information, he can do different things. Thus, if the user needs to join more one organization to the system, he can easily add one organization. For this is enough to click in the “Add Organization” button. Note that, if the user tries to add an

organization with the same name of some existent organization, the system will alert that already exist an organization with this name. On the other hand, the user also can remove an organization, in this situation occurs something similar like in the action "add organization". So if the administrator tries to remove an organization with data associated, what the platform does is alert the user and prevents this action.

Finally, the user to consult organization data, he needs to click in the "Edit" button. After click on this button, the platform presents the information associated with the respective organization. In the next sections we will describe the information presented and associated with the organization.

4.3.5.1. Users

In this section (), the platform shows users' information associated with the selected organization. Basically, an administrator in this page can do three different actions over users. The referred actions are add, remove and edit users. So, if an administrator wants to add a user, he just has to click on the "Add user" button and inserts the respective data. After the administrator inserts data, the platform will verify if all data are correct, in specific the platform checks if the user email is not repeated. Also, the administrator can edit user data or remove a specific user, but when he tries to do one of the referred actions the system verifies if the user is associated with some study in processing. If the user is associated with a processing study, the platform will prevent the respective action, in the way to ensure that the change does not affect the study. In the Figure 59, we show the described section.

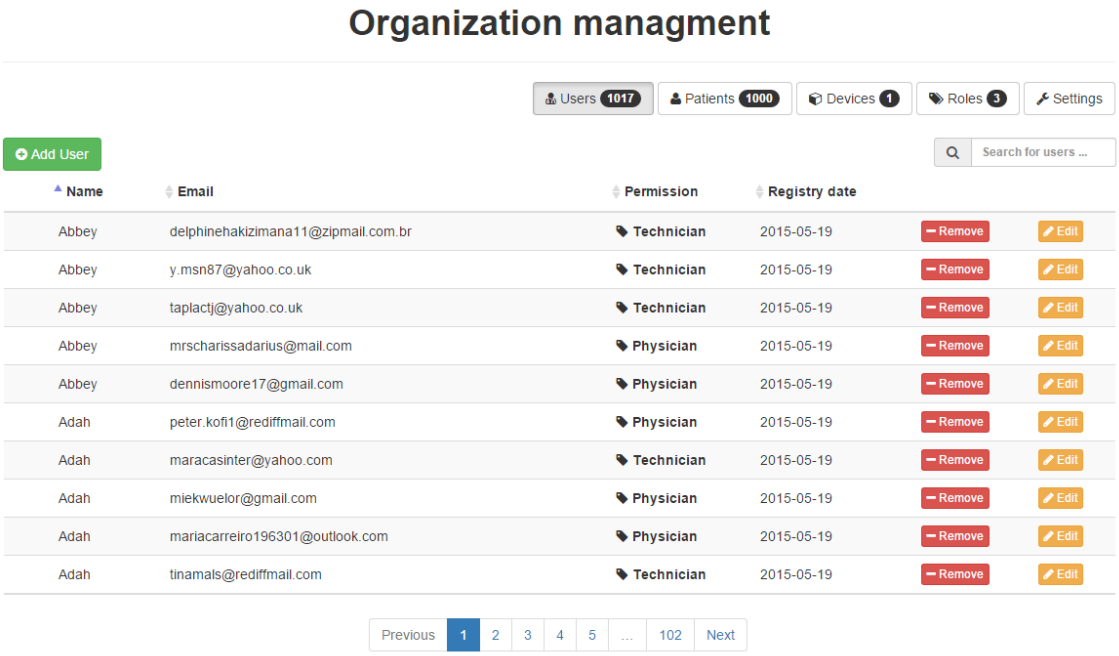

Figure 59 – Users section integrated in the management page

4.3.5.2. Patients

The patients' section (👤) will follow the same structure of the user section (👥), where we can do the same things. However, like the name expresses all the actions executed in this section are over the patients (Figure 60).

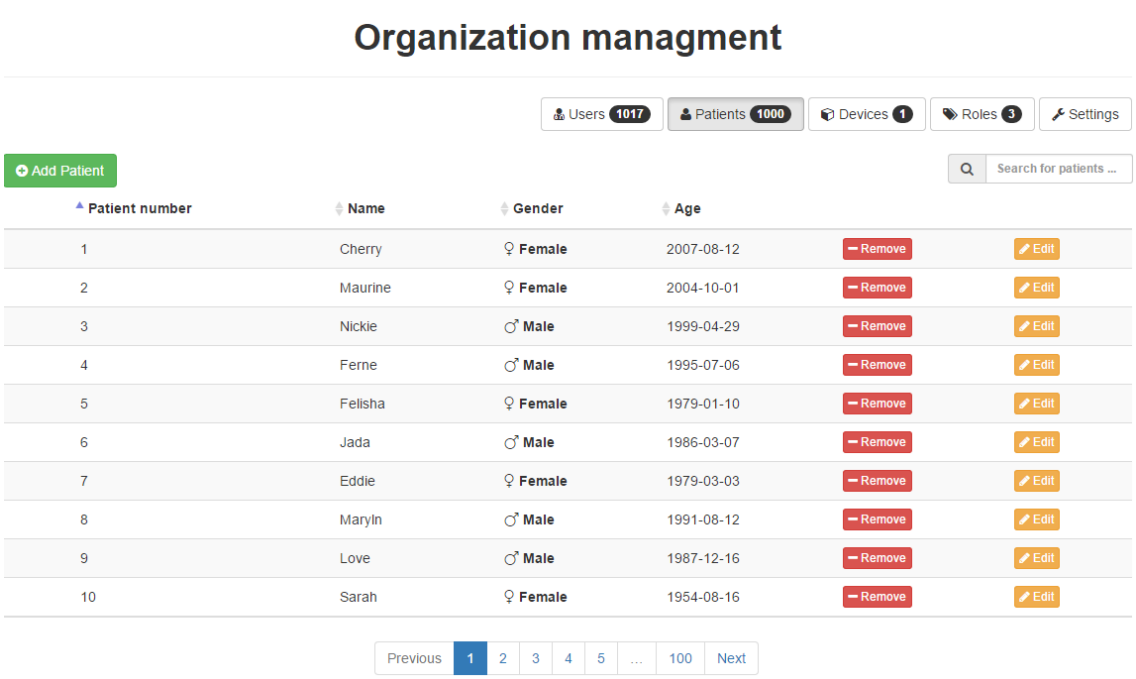

Figure 60 – Patients section integrated in the management page

4.3.5.3. Devices

How we described before in this document, one organization has associated one device. Considering that the organization has already one device, when the user tries to add another, the respective platform prevents this action. Here, also it is possible to edit and remove the respective device.

Note that, when a user adds a device, he had to verify if the same is connected, by clicking in the button Test Connection (Figure 61). Only if the device is connected with the platform and if all fields are correct is that the user can add the device.

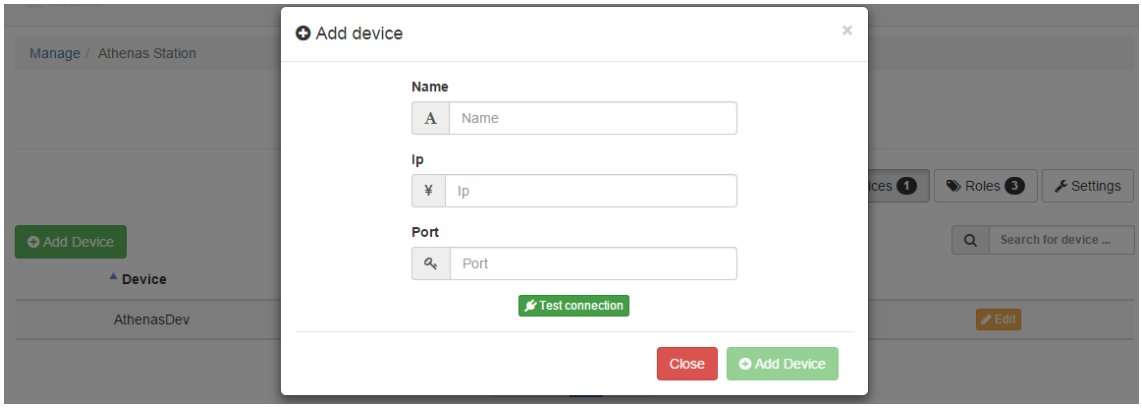

Figure 61 – Devices section integrated in the management page

4.3.5.4. Roles

The roles section (🏷) is where the administrator can define the several roles associated with each organization (Figure 62). Associated with each role the administrator can edit which permissions he wants to connect. It is in the definition of these roles that the administrator indicates what actions each user can do. For example, in the figure below, we see that the technician and the administrator role have the permission edit the device (✎). While the physician role does not have this permission. So, like we saw in the device page all users can visualize the information about the device (👁), but only the technician and the administrator have permissions to execute actions over the device (✎). Summarizing, the roles section is much important because it allows to define what each user with a specific role can do in the organization.

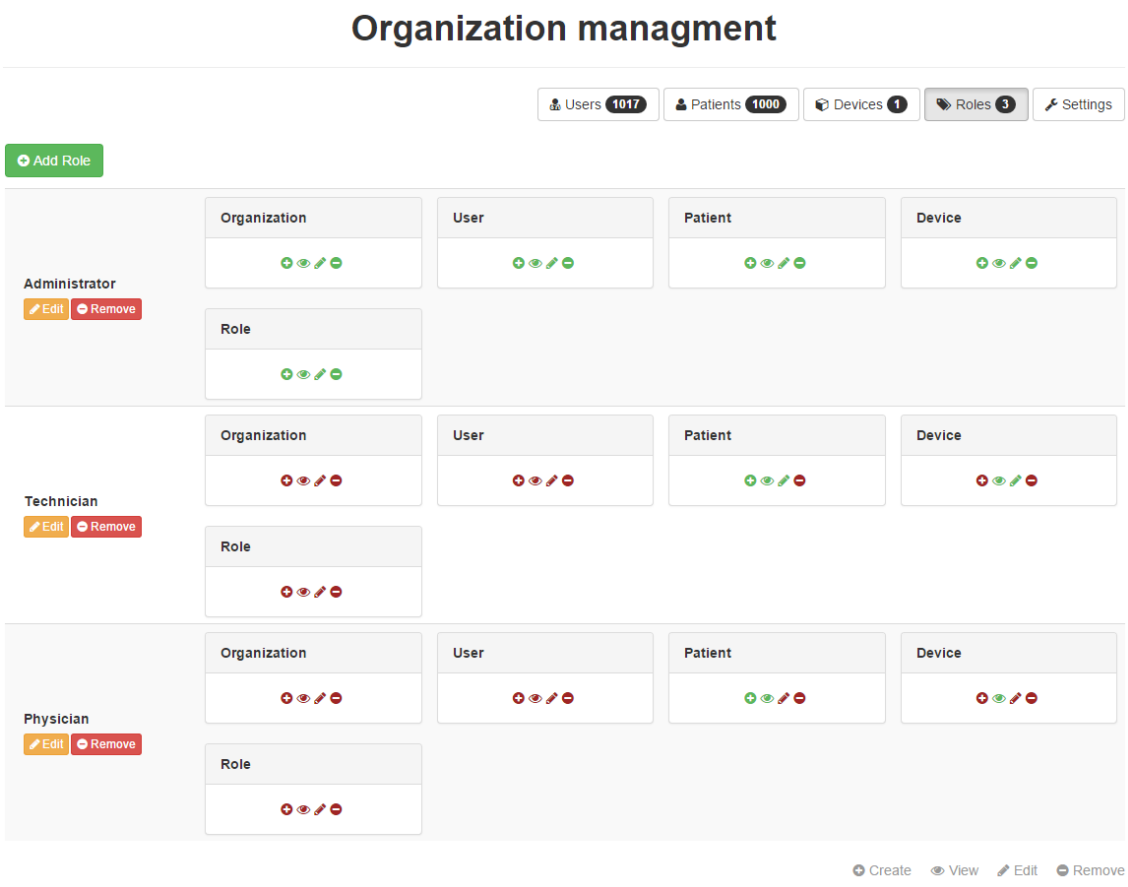

Figure 62 – Roles section integrated in the management page

4.3.5.5. Settings

The section settings (🔧) allows to see and edit data of the organization. So, when the administrator clicks in the "Settings" button, the platform

presents a modal (Figure 63) with the organization data and where the administrator can update these data.

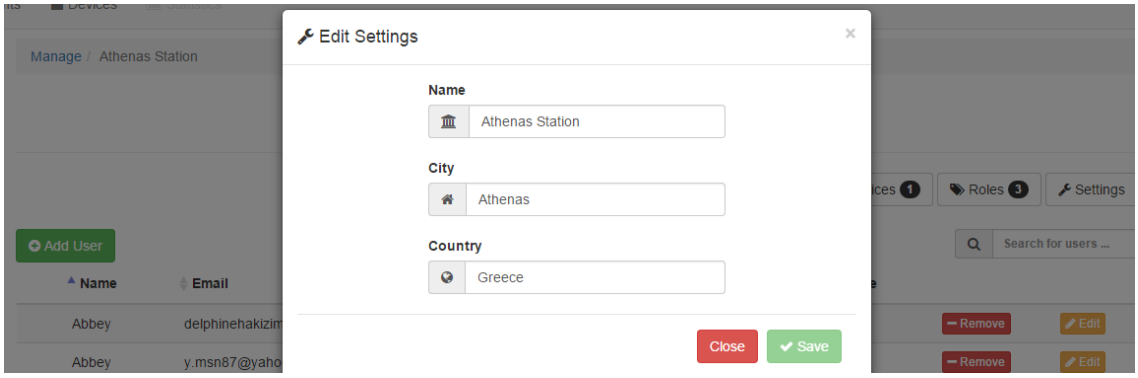

Figure 63 – Settings section integrated in the management page

4.4. Solution security

During the development of the platform, we always have in attention security and authentication factors. These factors are important in the platform because we need to assure that all the stored data and every connection between the server and the client are properly authenticated and secured. Another fact, it is that the platform has users with different roles, and each role has different permissions. In order to resolve all these problems, we implemented a role-based access control plugin to all the provided services. This way, we ensure that each method called from the client-side of the application is executed only if a user with the right permissions has requested the information, preventing security violation and assuring that the stored data are consistent and secure.

As we expressed and described in the last paragraph, to ensure a flexible and extensible security platform, we implement the RBAC plugin. This plugin only allows access to specifics web-services users with the correct role. This way, we needed to define the followed roles (Table 5):

- Administrator: this role represents all HemoSpec administrators. They have full access to the application and simultaneously they have access to all methods;
- Physician: this role is associated with users who are responsible for monitoring the patient process;

- Technician: assigned to all users who have the function of follow the patient analysis.

Table 5 – Different roles and respective permissions in HemoSpec Platform

Permission	Administrator	Technician	Physician
Add organization	✔	✘	✘
View organization	✔	✘	✘
Edit organization	✔	✘	✘
Remove organization	✔	✘	✘
Add user	✔	✘	✘
View user	✔	✘	✘
Edit user	✔	✘	✘
Remove user	✔	✘	✘
Add patient	✔	✔	✔
View patient	✔	✔	✔
Edit patient	✔	✔	✘
Remove patient	✔	✘	✘
Add device	✔	✘	✘
View device	✔	✔	✔
Edit device	✔	✔	✘
Remove device	✔	✘	✘
Add role	✔	✘	✘
View role	✔	✘	✘
Edit role	✔	✘	✘
Remove role	✔	✘	✘

In order to ensure proper authentication and authorization of the services, we assign the correspondent permission/permissions to each method. So, when a user accesses to the platform, the respective role is associated to him. Therefore, when a user invokes a method, the platform will verify if the user role contains some permission assigned in the method. If the platform verifies that the user has the right permission the requested is accepted, otherwise the requested is denied.

Summarizing, with the implemented plugin we ensure that each web service are robust and secure. These characteristics provide trustful services and avoid undesired requests. Moreover, the communication between the client and server is performed using a secured and encrypted channel using Hypertext Transfer Protocol Secure (HTTPS), in order to guarantee complete protection of the exchanged data.

Chapter 5

Evaluation

In this chapter, we will speak about the usability tests, heuristics evaluations and performance tests. Firstly, we will explain the importance of usability tests on the development of each platform. Also, we will explain the importance and the gains with the respective tests and we will present the results obtained from the realized usability tests. Secondly, we show the several heuristics covered during the heuristics evaluations and the several problems encountered. Finally, we present and analyze the performance results of the HemoSpec platform.

5.1. Introduction

The objectives of the usability tests and heuristics evaluations are to assess the design and test the system to ensure it behaves as expected. While, the performance tests demonstrate that the platform corresponds with the performance levels required. This way we need to evaluate the HemoSpec Platform. Evaluation is a development life cycle process and must be done according to principles and prototyping techniques because it is easier to change a design in the early stages of development that in later stages. This evaluation can be done by the designer or a usability expert (without final user interaction) and by a user who tests the actual use of the system [20].

One of the best and most popular ways to gather the maximum information is to observe users interacting with the system and analyzes the environment variables such as user's facial expressions, their comments or screen footage. To provide useful insight into problems within an interface

and if it meets user's requirements through their decision processes and attitude, we can use a Think Aloud method. This method collects users' description about what they are doing, what they believe is happening, the reason behind an action of a particular task and what they are trying to do [20]. This strengthens the usability tests providing large amounts of information from many sources that illustrate well the system debilities.

During the usability tests the tester observes users while they are using the system and find usability problems. However, can exist different types of tests defined by the development phase of the project, these tests can be done before development, during and after what affects the formality of the testing method [21].

Formality in testing is the type of methodology where users are interviewed about the system, how they use it, how they like it and which kind of software they use. Normally, this is done in front of a computer while users talk about what they are seeing and doing. The tester notes user reactions, comments and mistakes, giving the participant progressively harder tasks over the course of the session. According to Steve Krug [22] usability tests must be done because testing is better than no testing and with an iterative testing process, a live audience analysis and a representative group of users, we can find almost all the problems – Steve Krug also refers that it is more important having a small group of users with several tasks iterations than a lot of users and too few iterations.

Concluding, the importance of the several tests is to replan iteratively the current design and methodology, solving design and performance issues or to learn the weaknesses for the future releases [23].

5.2. Usability tests

The importance and the gains of the user testing led to a testing and usability plan (appendix A.1) to be carried out by students of the University of Aveiro. This plan asks each user to complete a series of tasks while they are being observed by a tester who notes all important comments along the

execution of the task, goal achievement, need for help, number of clicks and ease of execution.

Important and vital to the success of any user testing plan, users were selected with equal profile based on specific criteria to match the expected user population – similar age and level of education, experience with computers and technology domain [20].

When performing usability tests there are some problems that can be often found as users getting disoriented or confused about the user interface due to existing elements, the lack of them or if the functionality is not clear. The excess of information can also degrade the understanding of the user interface [22].

Keeping this in mind, the usability tests were performed to understand what was bad in the HemoSpec Platform, specifically where users failed during the test. It was also pretended to gather the maximum suggestions and observations about user experience during the test that could improve usability.

The test was made by 12 engineering students of computers and telematics that interacted directly with the HemoSpec Platform. During the several tests with the students, the tester had to register the user performance for each task:

- Number of clicks done;
- Completion of the task – boolean value;
- Mistakes made – quantitative by three ranges;
- Got lost – quantitative by three ranges;
- Call for help – boolean value;
- Difficulty felt by the user as judged by the tester;
- Relevant observations.

It was intended to gather relevant information about performance over each task, and if the users found each one easy and accessible or if they had problems with task completion. The need for help and disorientation with a probable high number of mistakes, tells if the task must be improved functionally or the interface elements that are used. Users were also asked to

evaluate a set of interface characteristics (appendix A.1) of HemoSpec Platform using a qualitative scale from 1 to 5 where 1 has a more negative value and 5 has a more positive value.

The results are illustrated by charts that use this scale.

5.2.1. Results

The users that test our platform having no experience with medical web platforms, which may affect the usability and functionality of the platform. However, the final questionnaire shows a positive reaction about the platform (Figure 64, 65).

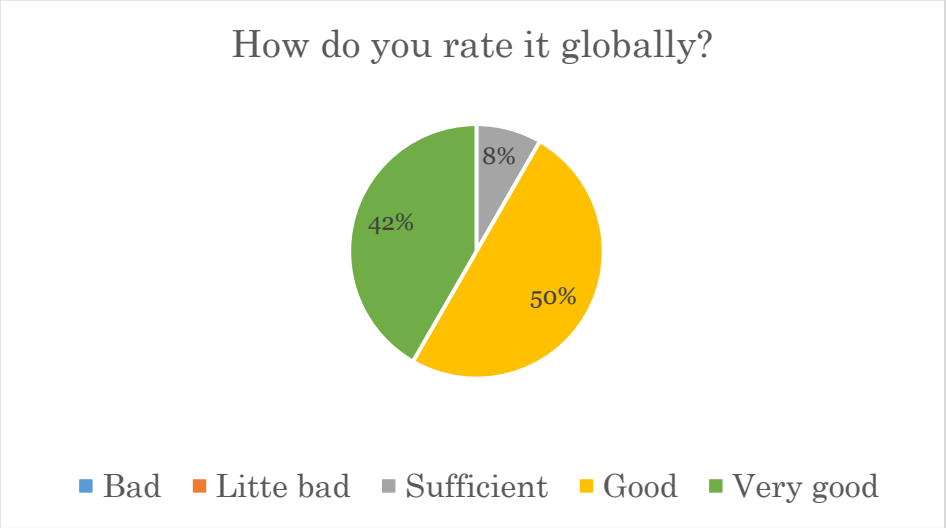

Figure 64 – Global classification of the platform

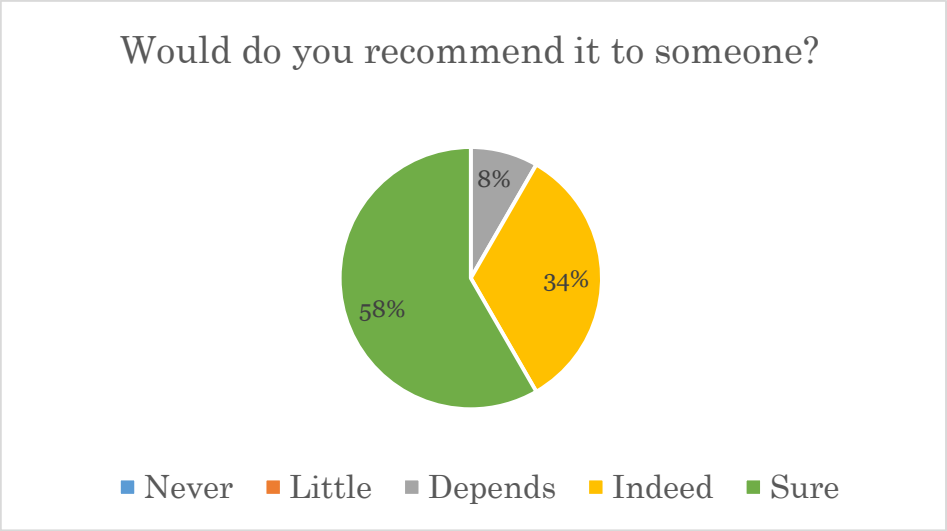

Figure 65 - Level of platform recommendation

In the Figure 66 we can analyze the difficulty rate of tasks realized during the usability tests. The level of difficulty of each task increase along of the test, with the objective to verify if the user interacting with the interface understands how the platform works. Observing the results we concluded that in general all tasks were easy of doing. However, to confirm our expectations, we asked to different users if they felt the tasks easy according with their evaluations. The users explained that initial tasks were easy instead of the last tasks, but with the use of the platform they felt more suited with the platform. Thus, even the last tasks make it accessible.

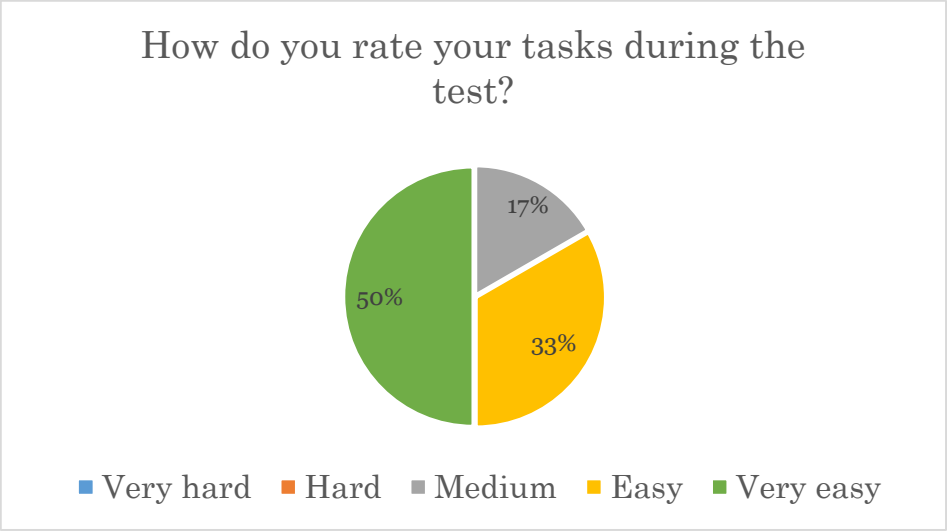

Figure 66 – Level of difficulty of the several tasks during the usability test

Based on the results presented in the Figure 67 and in the positive reviews received from the evaluators, we concluded that the platform design comply with the initially proposed objectives. In other words, we can classify the platform like an attractive and direct platform to the users.

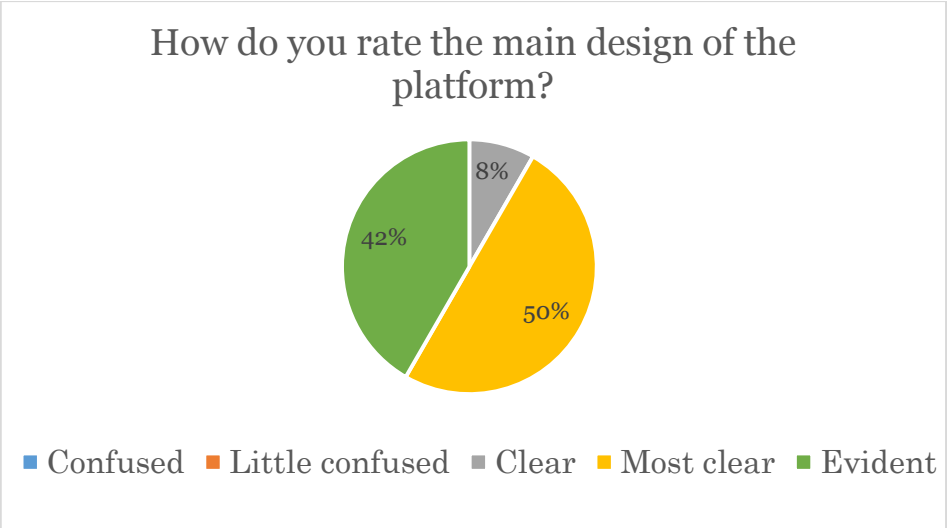

Figure 67 – Classification of the design of the platform

As mentioned, the platform has advantages in being attractive and direct for the users. However, to guarantee these characteristics also is necessary to ensure an intuitive navigation along the platform. In the Figure 68, we can see a chart that presents the opinion of each evaluator in relation with the browsing platform.

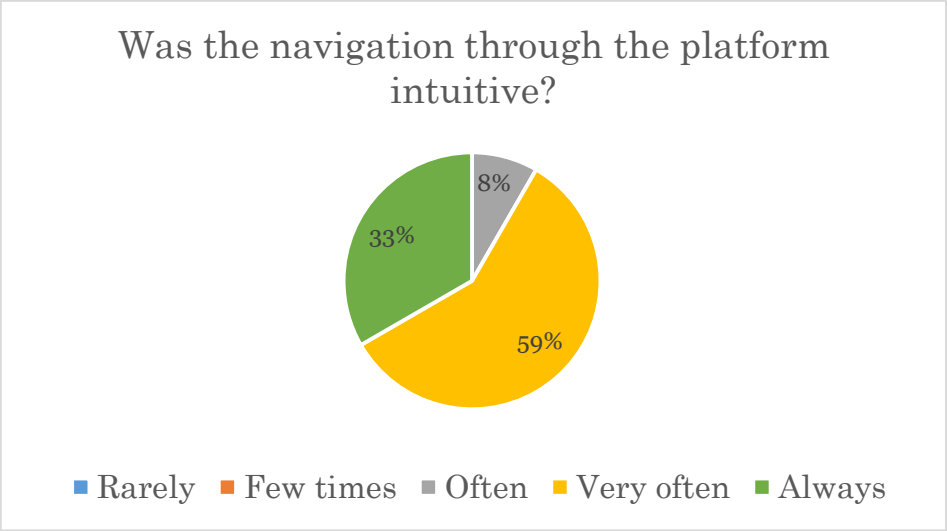

Figure 68 – Classification of the navigation in the HemoSpec Platform

Finally, we can visualize in the last figure of this section (Figure 69) a chart that shows the values about the difficulty that users felt in understand like the platform works.

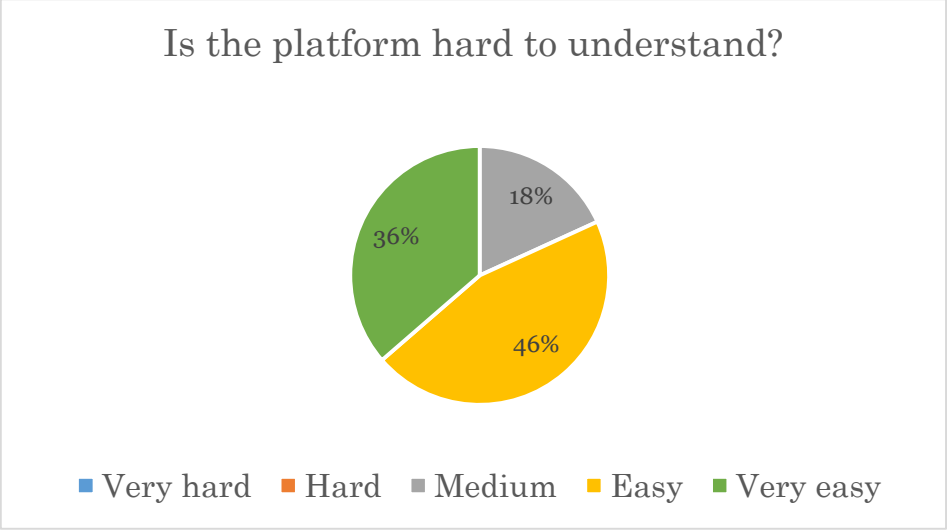

Figure 69 – Level of difficulty of understand the platform

As previously mentioned, usability tests do not enable us to find all the problems which is why heuristic evaluations often take place to cover some of encountered problems [24].

5.3. Heuristic evaluation

Usability tests are indeed important to find the major problems when the solution is still under development or at a beta stage. Other methods such as heuristic evaluations can be done in the early stage and in an existing design that detected problems can be fixed. Originally, heuristic evaluation was developed as a usability method for those who had some knowledge about usability principles not involving potential users in the tests [25].

According to Jeffries [26] and Desurvire [24], expert heuristic evaluators find more problems than any other evaluation technique, including usability testing [27]. This proves that evaluators trained in usability issues and heuristics principles find more problems than inexperienced users but these evaluation techniques are not accurate by themselves, and need to be used together: user's lack of experience and the naivete using a novel application proves to be better at finding unknown or minor problems besides the major ones.

A heuristic evaluation by trained evaluators was conducted to find the potential problems in the HemoSpec Platform, taking advantage of their skills and knowledge. This evaluation is based in the follow usability principles [28]:

- Visibility of system status;
- Match between system and real world;
- User control and freedom;
- Consistency and standards;
- Error prevention;
- Recognition rather than recall;
- Flexibility and efficiency of use;
- Aesthetic and minimalist design;
- Help users recognize, diagnose, and recover from errors;
- Help and documentation.

Heuristic evaluation is not set to fix usability problems or to provide a way to assess the probable quality of any redesign but to explain each observed problem against the principles listed. Severity ratings can be used to provide an estimate of the need for additional usability efforts, using

factors such as the frequency the problem occurs, its impact and persistence, but it is common to combine all of them into just one, to facilitate prioritizing, and decision-making [29]. Basically, it is a 4 level scale: 0 rating means the evaluator does not think there is any problem with the interface; 1 reflects a cosmetic problem; 2 a minor usability problem; 3 a major usability problem and 4 means a usability catastrophe.

The various problems encountered aim to highlight the weaknesses, evidencing improvements to the current user interface to become more efficient and user-friendly. The improvements ensure a cohesion between them and an operational flow that allows the user to use the application as a whole. The revision cycle of the design and implementation of these improvements should come in several iterations so that it is properly resolves and the integration of new functionality or services may be performed in a modular way and on a stable application. Let's take a look at the results of the heuristic evaluation.

5.3.1. Results

With the collaboration of 5 engineering students of computers and telematics as evaluators, with the basic knowledge and academic training for the heuristics evaluations, the results convey a positive reaction to HemoSpec Platform, with few cosmetic and minor usability problems.

Firstly, evaluators said that some of the buttons and labels should have more visibility, should be flashier, helping users to easily see the elements and the respective information associated with the element. Secondly, evaluators reinforced that will be beneficial in terms of platform perception, that when an action is associated with one justification (analysis restarts), the platform should advise users that have to write their decisions and only after they can execute the respective action.

Other 2 issues are related to the number of clicks associated with certain actions. In specific, when in the patients page technicians are alerted that there are request studies pending, the platform when detects clicks over the notification should automatically does a search over the patients in this status. Avoiding that users after verifying the notification still they have to use the search tool to find patients associated. The other problem is on the

details' consult page, when the analysis is in processing. If one of the modules changes to error or lost connection status should appear one button associated to the respective module. This button should redirect the user to the module's device page, avoiding in this way that the user has to click on button device to go to device page and after click on settings button of the respective module to go to module's device page.

5.4. Performance tests

A performance test can be realized in different types of applications, such as desktop or web applications. The same tests can be developed in several phases of the application development.

We will focus just in the performance tests oriented to the web applications that satisfy our necessities. These tests are divided into 3 main types:

- Load test – to verify application behavior under normal and peak load conditions;
- Stress test – to reveal application bugs that surface only under high load conditions;
- Capacity test – to determine how many users and/or transactions a given system will support.

According to a study done by the Nielsen Norman Group [30], the response times of the websites are divided into 3 different levels. Each level causes specific reactions in the users. The levels referred are as follows:

- To 0.1 seconds – the user has the feeling that the response is instantaneously;
- To 1 seconds – the system keeps users connected;
- From 10 seconds – users wish that the page load will be fast.

Nowadays, the performance of a site is mostly affected by the frontend and not the backend, like many people think. This is because all the data to be processed and presented to users is done in the frontend side. Note that 80-90% of the time spent by users waiting for pages to load is responsibility of the frontend [31].

Then, we will show the Figure 70, where we present a chart with the performance values of the platform. In our study, we use load tests since comply with our requirements, that evaluates the performance of the platform according to data growth.

5.4.1. Results

In the next chart, we present the response times that our platform delays to respond according to different number of patients. We compare 2 different requests, in the first request we just load patients' data and in the second request we load the actions associated with each patient analysis. The second request is more complex which tends to be slower.

After we visualize the chart that presents the values obtained from the performance tests, we can conclude directly that the respective chart is constituted by two sub-charts with different order of magnitude. Thus, the sub-chart represented between 0 and 1000 patients is of linear order of magnitude, while the other is of exponential order of magnitude. Let's go to analyze the referred sub-charts. The sub-chart of linear order of magnitude compared with the times presented by Nielsen Norman Group complies with the requirements set, since the load time is practically immediate maintaining the user focused on the executed workflow. Concerning with other sub-chart already cannot conclude the same, since the platform performance with a patient number more elevated tends to be worse, which means that users get tired of the time that the platform takes to load the data.

However, this platform is still in a development phase and it is integrated in a clinical project under study, where it is estimated that the number of patients in a short-medium term does not reach a high number, allowing comply with requirements set, while the platform is improved to the level of performance. On the other hand, as the clinic study will be elaborated in medium-sized hospitals also favors the situation described.

This way, we can conclude that the solution comply with the proposal requirements.

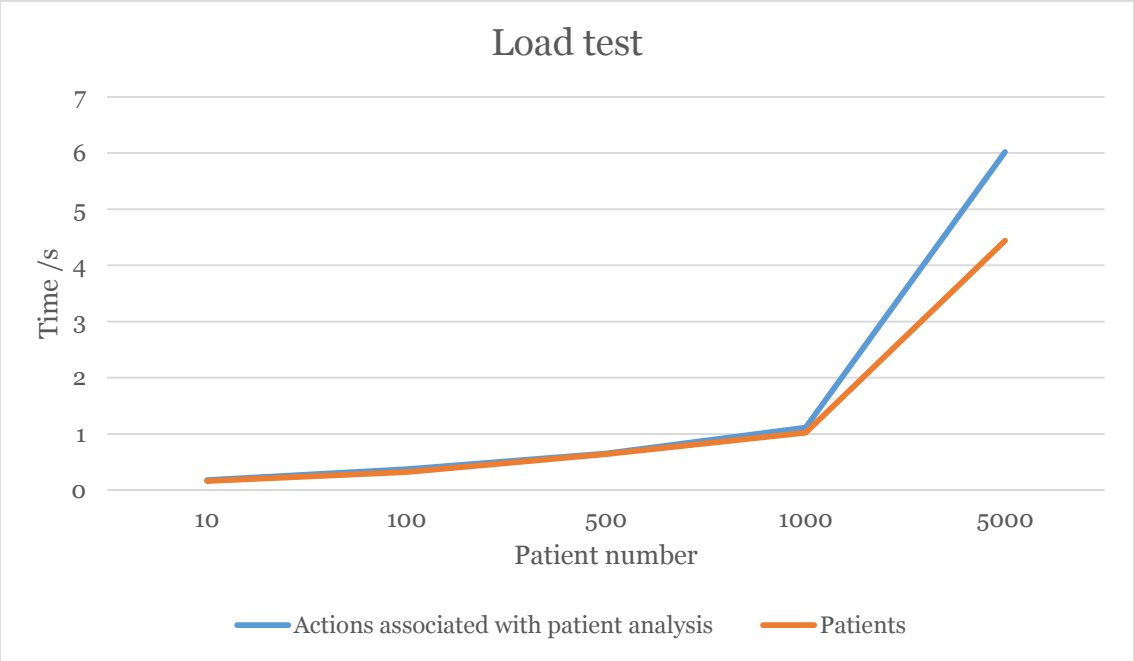

Figure 70 – Performance results of the HemoSpec Platform

Chapter 6

Conclusion

The main goal of this thesis was the study, design and development of a solution that covers the limitations and problems found in the current treatment process of sepsis disease. Thus, this work started with a careful requirements analysis in order to understand which features the solution should support. After getting the requirements of the solution we came up with an architecture for the desired solution. As a result we presented HemoSpec platform, a web-based platform that supports an automated diagnostic and monitoring of sepsis disease.

The user interface was carefully developed targeting simplicity and intuitive interactions, through direct visualization of the information associated with the patients, specifically the current consults status. Additionally, it provides a rich set of features to support the complete workflow of diagnostic and monitoring of sepsis in real - time, such as the control of a device associated with an organization, the control of the analysis process and the control of organizational data.

The developed solution intends, also, to reduce the time of diagnostic and monitoring of sepsis, thusly improving the rate of success of the treatment and reducing the mortality rate. For this aim, our solution needs to control just one unique analysis executed by the HemoSpec device, which ensures liability and data exchange security. Simultaneously, the proposed workflow speeds up the execution velocity of the respective analysis. To guarantee a correct processing, it was necessary to implement a communication protocol with the objective of ensuring that user actions and the data exchange are

instantaneous during the workflow without any occurring problems and respecting the proposed requirements.

In the way to evaluate the usability and performance of the HemoSpec platform, we realized specific tests with the objective of obtaining conclusive results. After the several tests were realized, the platform presented convincing results in terms of usability, liability and performance, exceeding expectation proposals. We can affirm that the developed solution meets the solution's goals.

6.1. Future Work

Though HemoSpec platform provides a rich set of features that makes it an innovative and flexible solution with many advantages for the sepsis treatment, we are aware that many features can be integrated in the platform to further improve it, such as:

- Improve the performance of the platform, in the way to support large increases in the number of patients and maintaining good performance levels;
- Integrate statistics section, which shows data about the current situation of the sepsis disease;
- Import additional and relevant documents to the study

Appendix A

User Interface Evaluation

This appendix contains all the documentation used to support the user interface usability evaluation, and the results obtained after a careful analysis.

A.1 Usability Test

A.1.1 Documents

Documents used in the usability tests.

hemospec

Questionnaire guide

Introduction and goals

The purpose of this round of testing involves feedback from the user to the proposed structure for the Master thesis in HemoSpec Platform, identifying serious problems during a development phase.

Specific questions to be answered:

- The navigation conventions make sense?
- The navigation structure is easy and intuitive?
- The desired information is easy to find?
- User knows, every time, where and what he can do?

User characteristics

- Age group representative: participants above 20 years;
- It is not fundamental representative genus division;
- Participants can in no way be connected to the development of the platform;
- Participants should be agile enough regarding computer usage.

Methodology

We will submit to each participant how the test will be executed telling them to speak out loud about everything they are seeing, doing and thinking during the session. If applicable, each participant will be called to fulfill a confidential form provided by us. Participants will then have a number of tasks to perform. Each session will have an approximated duration of twenty minutes.

Once all tasks done, it will be given to the user the possibility of placing questions. At last, it will be request each participant to fulfill a small inquiry. The test is presented in next page.

Test

- Welcome
- Presentation of the test
- Task lists:

    - **Task 1**
        - Login in the platform, as a technician. After, search for the patient Jacques Bolton and click in the assign button;
    - **Task 2**
        - To continue the workflow, start the patient's analysis;
    - **Task 3**
        - After the microfluidic separation module finishes the respective analysis, type some notes about these results in the module device page;
    - **Task 4**
        - Go to device page, stop and start the fluorescence module which has triggered an error;
    - **Task 5**
        - Go to patient page and verify if all data is ok;
    - **Task 6**
        - Verify the patient's consult's details and validate results;
    - **Task 7**
        - Logout from the platform and after login in the platform, as physician;
    - **Task 8**
        - Verify the current information and access to the consult's details of the patient's Jacques Bolton;
    - **Task 9**
        - Check for comments associated with the microfluidic separation module results;
    - **Task 10**

        - Check for technician answers in the technician validation section;
    - **Task 11**
        - Submit a treatment guide for Jacques Bolton medical issue;
    - **Task 12**
        - Add a new patient with the health number 60149 or higher;
    - **Task 13**
        - Request a study to the added patient;
    - **Task 14**
        - Logout from the platform and after login in the platform, as technician. Click in the assign button associated with the new patient;
    - **Task 15**
        - Start the patient analysis;
    - **Task 16**
        - Click in reanalysis button;
    - **Task 17**
        - Validate results;
    - **Task 18**
        - Logout from the platform and after login in the platform, as physician. Submit a treatment guide for the last patient analysis;
    - **Task 19**
        - Go to device page; search the history table for the patient that you added;
    - **Task 20**
        - Logout from the platform and after login in the platform, as administrator.
    - **Task 21**
        - Access to the manage page;
    - **Task 22**
        - Choose the Jena organization;
    - **Task 23**

        - Access to patient section and edit the name of the patient that you added;
    - **Task 24**
        - Change the organization name;
    - **Task 25**
        - Logout from the platform and login in the platform, as technician or physician;
    - **Task 26**
        - Go to device page; search in the history table for the patient added;
    - **Task 27**
        - Search again by the new name;
    - **Task 28**
        - Logout from the web platform.

- After test inquiry.

Tester guide

Here, we present a set of relevant patterns that the tester will try to follow during the writing of reports:

- Does the user understand well the objective for each task and respective interactions?
- Did he/she find easily the interface components that fit the purpose? Which was the location of the components he/she thought?
- How did he interpret each task? This means what he thinks he has to do, where to click, what action is under the button event and what is going to happen in consequence.
- Did the user need help to perform some task? What was the task and why did he/she get stuck?
- Did the user gain easily sensibility and experience using the platform?

During the writing of the reports, the tester will cover the maximum possible points discussed above without disturbing the user nor interfering with the test.

Report

Note: Use this space for additional notes or comments

User name:		Evaluation table of the tester									
Task	Number of clicks	Did the user finish the task?	Did the user make mistakes?	Got lost?	Call for help?	How easily observed 1 – Not easy 5 – Extremely easy					
1		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
2		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
3		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
4		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
5		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
6		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
7		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
8		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
9		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
10		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
11		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
12		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
13		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
14		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
15		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
16		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
17		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
18		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
19		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
20		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
21		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
22		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
23		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
24		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
25		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
26		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
27		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	
28		Yes: ___ No: ___	No: ___ Few: ___ Many: ___	No: ___ Little: ___ Very: ___	Yes: ___ No ___ Which?	1	2	3	4	5	

Script of usability testing

Introduction and goals

Hello, I am Tiago Lourenço and you will work with my project in this section. I'd like to start presenting you briefly what we are going to do and what I am trying to obtain.

So, today I will evaluate the structure of the HemoSpec Platform. Your experience will help me assuring that this platform is simpler, user-friendly and intuitive. During the session, there will be a tester controlling you and observing you while you explore the platform.

Please, remember I don't intend somewhat to evaluate you. I am just trying to understand how people use the platform. Do your best but don't worry with the results. Any question or problem along the essay you must tell, in order to develop a better product.

This way you will have a list of tasks that must be done in a current and beta version of the platform, therefore pay attention for every detail.

I ask you that, at each moment and per task, to speak out loud about everything you see and think, what you are looking for, what you are doing and what you expect to obtain. You will be under supervision of a tester who will note down main aspects of your behavior using the system, yours errors, tries and comments you make. The more information you provide the better.

Now, the tester will explain in general how works the platform.

After the explanation, if do you have any doubt ask? If not, lets begin.

Tasks list

To access in the platform use:

- Physician email: cjmlourenco@gmail.com
- Technician email: loirovila@gmail.com
- Administrator email: tiago.vf.lourenco@gmail.com

All users have the same password: Tiago9

Tasks list:

- **Task 1**
  - Login in the platform, as a technician. After, search for the patient Jacques Bolton and click in the assign button;
- **Task 2**
  - To continue the workflow, start the patient's analysis;
- **Task 3**
  - After the microfluidic separation module finishes the respective analysis, type some notes about these results in the module device page;
- **Task 4**
  - Go to device page, stop and start the fluorescence module which has triggered an error;
- **Task 5**
  - Go to patient page and verify if all data is ok;
- **Task 6**
  - Verify the patient's consult's details and validate results;
- **Task 7**
  - Logout from the platform and after login in the platform, as physician;
- **Task 8**
  - Verify the current information and access to the consult's details of the patient's Jacques Bolton;
- **Task 9**

        - Check for comments associated with the microfluidic separation module results;
    - **Task 10**
        - Check for technician answers in the technician validation section;
    - **Task 11**
        - Submit a treatment guide for Jacques Bolton medical issue;
    - **Task 12**
        - Add a new patient with the health number 60149 or higher;
    - **Task 13**
        - Request a study to the added patient;
    - **Task 14**
        - Logout from the platform and after login in the platform, as technician. Click in the assign button associated with the new patient;
    - **Task 15**
        - Start the patient analysis;
    - **Task 16**
        - Click in reanalysis button;
    - **Task 17**
        - Validate results;
    - **Task 18**
        - Logout from the platform and after login in the platform, as physician. Submit a treatment guide for the last patient analysis;
    - **Task 19**
        - Go to device page; search the history table for the patient that you added;
    - **Task 20**
        - Logout from the platform and after login in the platform, as administrator.
    - **Task 21**
        - Access to the manage page;

- **Task 22**
  - Choose the Jena organization;
- **Task 23**
  - Access to patient section and edit the name of the patient that you added;
- **Task 24**
  - Change the organization name;
- **Task 25**
  - Logout from the platform and login in the platform, as technician or physician;
- **Task 26**
  - Go to device page; search in the history table for the patient added;
- **Task 27**
  - Search again by the new name;
- **Task 28**
  - Logout from the web platform.

After test inquiry

Please help me fulfilling this inquiry based on your experience during the test.

How do you rate your tasks during the test?	Hard	1 2 3 4 5	Easy
How do you rate the main design of the platform?	Confused	1 2 3 4 5	Evident
Was the navigation through the platform intuitive?	Rarely	1 2 3 4 5	Always
Is the platform hard to understand?	Hard	1 2 3 4 5	Easy
How do you rate it globally?	Bad	1 2 3 4 5	Good
Would do you recommend it to someone?	Never	1 2 3 4 5	Sure

Bibliography

[1] H. Project, "Collaborative project," HemoSpec Project, Brussels, 2014.

[2] J. Gube, "7 Best Practices for Improving Your Website's Usability," Marshable, 12 09 2011. [Online]. Available: http://mashable.com/2011/09/12/website-usability-tips.

[3] N. Bevan, "Usability is Quality of Use," in *6th International Conference on Human Computer Interaction*, Yokohama, 1995.

[4] G. C. Platform, "Building Scalable and Resilient Web Applications on Google Cloud Platform," Google, 22 May 2015. [Online]. Available: https://cloud.google.com/solutions/scalable-and-resilient-apps.

[5] R. M. Kim Hamilton, Learning Uml 2.0, O'Reilly, 2006.

[6] E. Foundation, "Concept: Use-Case Model," Eclipse Foundation, [Online]. Available: http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/concepts/use_case_model_CD178AF9.html.

[7] C. M. community, "UML Use Case Diagrams: Tips and FAQ," Carnegie Mellon community, [Online]. Available: https://www.andrew.cmu.edu/course/90-754/umlucdfaq.html.

[8] SourceMaking, "Activity Diagrams," SourceMaking, [Online]. Available: https://sourcemaking.com/uml/modeling-business-systems/external-view/activity-diagrams.

[9] J. Dotson, "HTTP Vs. HTTPS: What's The Difference?," BizTech, 5 July 2007. [Online]. Available: http://www.biztechmagazine.com/article/2007/07/http-vs-https.

[10] J. Smith, "Desktop Applications Vs. Web Applications," Streetdirectory, [Online]. Available: http://www.streetdirectory.com/travel_guide/114448/programming/desktop_applications_vs_web_applications.html.

[11] J. B. Smelcer, "Desktop vs Web Application: Business or Technical Decision?," Fairfield Professionals, LLC, [Online]. Available: http://fairfieldprofessionals.com/index.php/insights/88-insights/125-desktop-vs-web-application-business-or-technical-decision.

[12] "Operações Assíncronas, Deferred e Promises," Abequar.net, 20 March 2013. [Online]. Available: http://www.abequar.net/posts/opera%C3%A7%C3%B5es-ass%C3%ADncronas,-deferred-e-promises.

[13] M. Bynens, "Pimp vs bluebird vs Q vs rsvp vs then promise," 27 May 2014. [Online]. Available: http://jsperf.com/pimp-vs-bluebird-vs-q-vs-rsvp.

[14] P. Antonov, "bluebird," GitHub, 1 September 2013. [Online]. Available: https://github.com/petkaantonov/bluebird/blob/master/API.md#thenfunction-fulfilledhandler--function-rejectedhandler----promise.

[15] J. Atwood, "Understanding Model-View-Controller," Coding Horror, 5 May 2008. [Online]. Available: http://blog.codinghorror.com/understanding-model-view-controller/.

[16] http://c2.com/, "Model View Controller," 20 November 2014. [Online]. Available: http://c2.com/cgi/wiki?ModelViewController.

[17] R. Nimesh, "Understanding RequireJS for Effective JavaScript Module Loading," sitepoint, 7 January 2013. [Online]. Available: http://www.sitepoint.com/understanding-requirejs-for-effective-javascript-module-loading/.

[18] T. Fredrich, "RESTful Service Best Practices," Pearson eCollege, 2013.

[19] N. I. Corporation, "Application Design Patterns: Master/Slave," National Instruments Corporation, 13 December 2006. [Online]. Available: http://www.ni.com/white-paper/3022/en/.

[20] A. Dix, J. Finlay, G. Abowd and R. Beale, Human-computer Interaction, upper saddle river: Prentice hall, 1997.

[21] J. Johnson, GUI Bloopers 2.0, Second Edition: Common User Interface Design Don'ts and Dos, San Francisco: Morgan Kaufmann, 2007.

[22] S. Krug, Don'T Make Me Think: A Common Sense Approach to the Web, berkeley: New Riders, 2006.

[23] J. Tidwell, Designing interfaces, Sebastopol: O'Reilly Media, 2010.

[24] H. Desurvire, J. Kondziela and M. E. Atwood, "What is gained and lost when using methods other than empirical testing," in *Conference on Human Factors in Computing Systems*, New York, 1992.

[25] J. Nielsen, "Finding usability problems through heuristic evaluation," in *Conference on Human Factors in Computing Systems*, New York, 1992.

[26] R. Jeffries, J. R. Miller, C. Wharton and K. M. Uyeda, "User interface evaluation in the real world: A comparison of four techniques," in *Conference on Human Factors in Computing Systems*, New York, 1991.

[27] R. Jeffries and H. Desurvire, "Usability testing vs. heuristic evaluation: Was there a contest?," in *Bulletin*, 1992.

[28] J. Nielsen, "How to conduct a heuristic evaluation," Nielsen Norman Group, 1 January 1995. [Online]. Available: http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/.

[29] J. Nielsen, "Severity Ratings for Usability Problems," Nielsen Norman Group, 1 January 1995. [Online]. Available: http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/.

[30] J. Nielsen, "Website Response Times," Nielsen Norman Group, 21 June 2010. [Online]. Available: http://www.nngroup.com/articles/website-response-times/.

[31] S. Souders, "14 Rules for Faster-Loading Web Sites," High Performance Web Sites, [Online]. Available: http://stevesouders.com/hpws/.