SafeCloud

Prototype, Maxdata SafeCloud-based healthcare platform

D5.4

Project reference no. 653884

August 2017

European Commission

Horizon 2020
European Union funding
for Research & Innovation

Document information

Scheduled delivery	31.08.2017
Actual delivery	31.08.2017
Version	1.0
Responsible Partner	Maxdata Software, S.A.

Dissemination level

Public

Revision history

Date	Editor	Status	Version	Changes
12.06.2017	P.Sousa	Draft	0.1	Initial version, section 2 done
13.06.2017	P.Sousa	Draft	0.2	Section 3.2 done
28.07.2017	P.Sousa	Draft	0.3	Only section 3.2.2 is missing
01.08.2017	F.Maia	Draft	0.4	Input from INESC TEC
01.08.2017	B.Ferreira	Draft	0.5	Ready for 1st internal review
16.08.2017	B.Ferreira	Draft	0.6	Added contributions and feedback from CYBER and INESC TEC
19.08.2017	P.Sousa	Final	1.0	Final release

Contributors

Bruno Ferreira (Maxdata)
Paulo Sousa (Maxdata)
Francisco Maia (INESC TEC)
João Paulo (INESC TEC)
Rogério Pontes (INESC TEC)
Ricardo Macedo (INESC TEC)

Internal reviewers

Karl Tarbe (CYBER)
Francisco Maia (INESC TEC)
João Paulo (INESC TEC)

Acknowledgements

This project is partially funded by the European Commission Horizon 2020 work programme under grant agreement no. 653884.

More information

Additional information and public deliverables of SafeCloud can be found at http://www.safecloud-project.eu

Glossary of acronyms

Acronym	Definition
EU	European Union
SQL	Structured Query Language
ORM	Object-relational mapping
SaaS	Software as a Service
BLOB	Binary Large OBject

Table of contents

Executive summary

This deliverable describes a first prototype of the Maxdata SafeCloud-based healthcare platform. Main topics presented:

- An overview of the CLINIdATA® eHealth Solution, Maxdata's most important product.
- A description of the first version of the prototype that covers 2 of the 3 scenarios presented in D5.2 (SaaS and hybrid deployments), including the challenges that were addressed during the development of the integration between CLINIdATA® and the Secure SQL Engine.
- A set of demonstration procedures that show in practice the implementation of a use-case described in Deliverable D5.2.

1 Introduction

The objective of work package 5 (WP5) is to deploy SafeCloud in order to provide its novel privacy features to new and improved products and services. Two main use cases will be demonstrated: a cloud storage platform and a healthcare platform.

In deliverable D5.2 we presented the design and requirements of the healthcare platform that will be built by Maxdata on top of the SafeCloud framework.

This deliverable (D5.4) describes a first prototype of the Maxdata SafeCloud-based healthcare platform.

The deliverable is structured as follows.

- Section 2 presents the design of the healthcare platform. This platform is based on CLINIdATA®, an eHealth web application owned by Maxdata.
- Section 3 describes the different components of the prototype v1 and the challenges faced during its integration.
- Section 0 describes the healthcare laboratory information system demonstrator, making use of the SafeCloud Secure Queries Layer.
- Conclusions about the prototype are summarized in Chapter 5.

2 CLINIdATA® eHealth Solution

As described in deliverable D5.2, the healthcare platform that is being built in SafeCloud, is based on CLINIdATA®, an eHealth web application owned by Maxdata.

CLINIdATA® is composed of several modules, including a healthcare laboratory information system and an epidemiological surveillance system dedicated to supporting Hospital Infection Control Committees.

2.1 Product context

CLINIdATA® addresses different types of clinical and non-clinical organizations. In the clinical domain, CLINIdATA® is used by various healthcare organizations, including hospitals, clinics, laboratories, and primary care units. In the non-clinical domain, CLINIdATA® is used in laboratories of many areas, including water, food, and environmental health.

Maxdata current approach is to install its software on customer premises. This approach has allowed Maxdata to conquer most of the Portuguese market - currently present in more than 80% of national public hospitals -, but now the company wants to sell its products abroad and one of the ways is to deploy them on the cloud and sell them as a service (SaaS).

Architecture-wise, CLINIdATA® is a web application composed of 3 layers:

- **Presentation layer**: Includes the presentation logic and runs on any common browser (e.g., Google Chrome, Mozilla Firefox, Microsoft Internet Explorer). HTML and JavaScript code is generated automatically using the Google Web Toolkit[1];
- **Business logic layer**: Set of services, running on the server side, that implement the business logic. These services are implemented in Java using the Spring Framework[2];
- **Database access layer**: Set of methods, running on the server side, used to access the relational database where data is persisted. These methods use the Hibernate ORM Framework[3] to abstract the concrete DBMS (e.g., Oracle[4], PostgreSQL[5]) being used.

CLINIdATA® is used by different types of organizations, ranging from small laboratories with a few dozens of professionals and hundreds of transactions per day, to very large hospital clusters with thousands of professionals and tens of millions of transactions per day. Deliverable D5.2 details the typical workloads of large versus small organizations in terms of users, analyzers, exams, storage, and database transactions.

2.2 Product capabilities/functions

CLINIdATA® is an eHealth web application that includes the following main features:

- 100% web application - no plugins required on the client side, only a web browser is need to run the application;

[1] http://www.gwtproject.org

[2] https://projects.spring.io/spring-framework

[3] http://hibernate.org/orm/

[4] https://www.oracle.com/database/index.html

[5] http://www.postgresql.org

- Cross-platform application where server components may run on any common operating system (e.g., Linux, Mac OS X, Solaris, Windows) and relational database (e.g., MySQL, PostgreSQL, Oracle, SQL Server);
- Overall management and control of clinical laboratories, including technical and financial aspects of all areas of Clinical Pathology (e.g., clinical chemistry, hematopathology, microbiology, immunology) and Anatomic Pathology (e.g., histopathology, cytopathology, immunohistochemistry);
- Overall management and control of different types of non-clinical laboratories, e.g., water, air, food;
- Management of the entire exam process in clinical and non-clinical laboratories, including the 3 typical phases:
    - **Pre-analytical**: Prescription, specimen collection, specimen transport;
    - **Analytical**: Exam realization, quality control, results validation/approval;
    - **Post-analytical**: Access to exam results by patients & clinicians, billing.
- Epidemiological surveillance tool dedicated to supporting Hospital Infection Control Committees in the prevention, identification and monitoring of infections;
- Real-time interface that supports more than 300 different automated analyzers;
- Integration with dozens of other clinical and non-clinical information systems (e.g., intensive care unit, patient identification, billing, regional health portals);
- Rules engine, allowing the incorporation of intelligence and safety throughout the whole application;
- Business Intelligence (BI) platform that allows to view and explore statistics through dashboards and other visual elements.

2.3 Constraints

When deployed in healthcare organizations, CLINIdATA® manages personal data, so it should be compliant with regulations on personal data protection (e.g., in Europe - General Data Protection Regulation[6], in the USA - Health Insurance Portability and Accountability Act[7]).

2.4 Cloud Deployment Scenarios

Taking advantage of the benefits of the SafeCloud framework, CLINIdATA® may be deployed on the cloud in 3 different scenarios depending on the customer type and on the features that the customer wants to access:

- **SaaS deployment**: for small and medium-scale healthcare organizations that want to reduce costs on infrastructure, CLINIdATA® will be offered using the software-as-a-service (SaaS) model where all components are deployed on cloud providers contracted by Maxdata.
- **Hybrid deployment**: for large healthcare organizations that want to reduce costs on infrastructure but do not trust any cloud provider, CLINIdATA® computation/processing will be installed on customer's premises making use of

[6] http://ec.europa.eu/justice/data-protection/

[7] http://www.dhcs.ca.gov/formsandpubs/laws/hipaa/Pages/1.00WhatisHIPAA.aspx

SafeCloud components to access data securely stored on untrusted cloud providers contracted by healthcare organizations.

- **Analytics deployment**: for groups of healthcare organizations that want to share analytics on their combined data without revealing the private data of each organization, CLINIdATA® computation/processing will be installed on each of the involved customer's premises making use of SafeCloud components to access data securely stored on untrusted cloud providers – contracted by the healthcare organizations - in a way that each healthcare organization is only allowed to put in its private data but cannot make direct queries for data – i.e. only aggregated queries are possible.

Deliverable D5.2 describes each of these cloud deployment scenarios in more detail, including the way CLINIdATA® will take advantage of the SafeCloud framework to achieve its goals.

3 Prototype v1

3.1 Introduction

The first version of the healthcare platform prototype covers the SaaS and hybrid deployment scenarios described in D5.2 and summarized in Section 2.4 of this deliverable. Figures 1 and 2 present how each of the deployments was instantiated in this first version of the prototype.

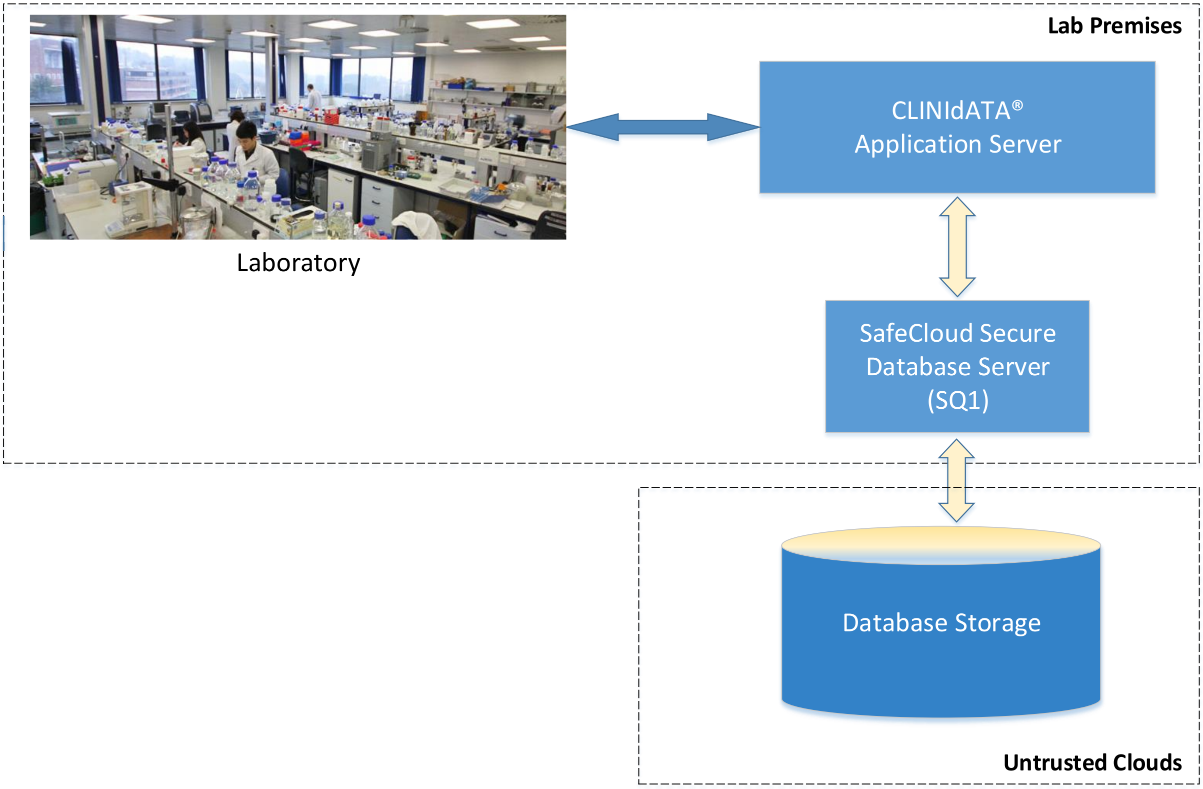


Figure 1 – First version of the healthcare platform: SaaS deployment

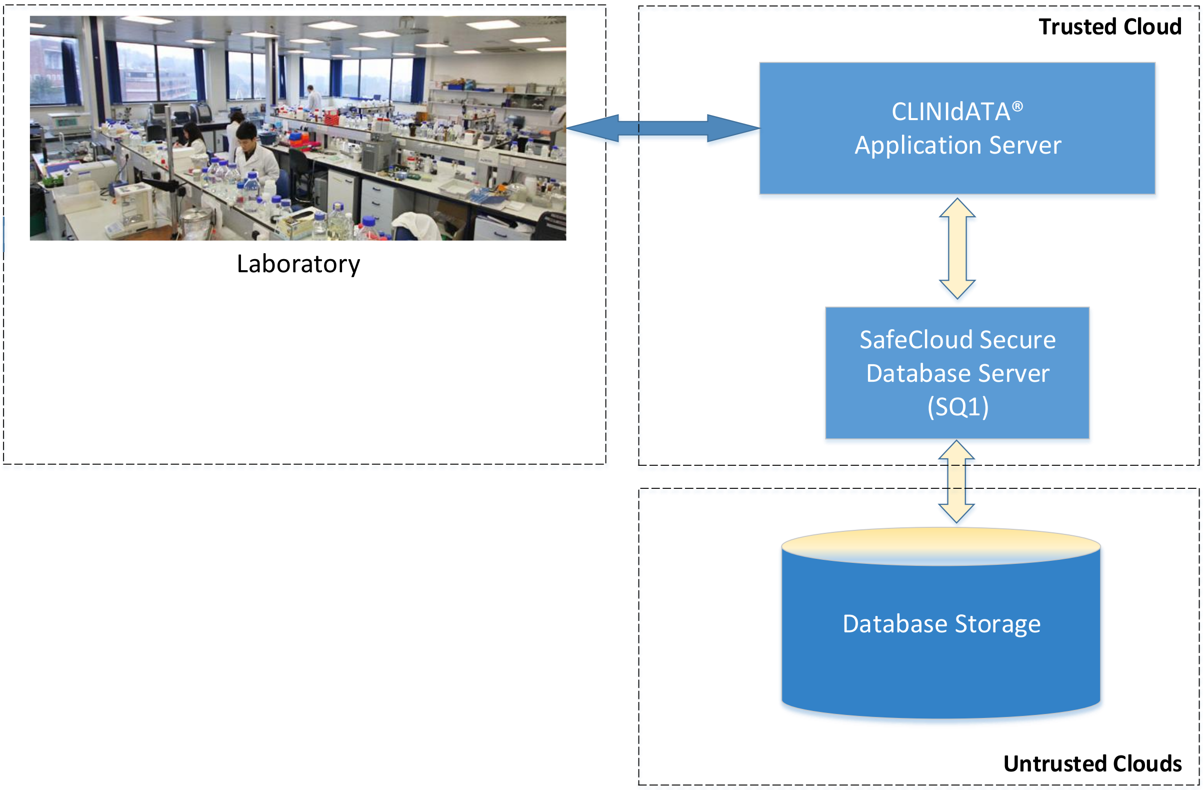


Figure 2 - First version of the healthcare platform: hybrid deployment

The main work done was the integration between CLINIdATA® and solution 1 of the SafeCloud Secure Queries Layer (Secure database server - SQ1). The details of this integration are presented in the following sections.

3.2 Integration of CLINIdATA® with SafeCloud components

3.2.1 CLINIdATA® Integration Work

In order to integrate CLINIdATA® with solution 1 of the SafeCloud Secure Queries Layer (Secure database server - SQ1), some changes had to be done on the database access layer of the CLINIdATA® web application to fully support the Apache Derby[8] SQL interface provided by SafeCloud SQ1. However, these required changes were greatly eased thanks to the use of the Hibernate ORM Framework on the CLINIdATA® database access layer, as it was described on the Section 2.1 of this document. The use of this framework allowed to reuse almost every SQL query that CLINIdATA® is already using on top of other database management systems that are supported since the beginning, namely Oracle and PostgreSQL. Nevertheless, some minor adjustments had to be done on some specific queries, for instance the queries used to validate the application licenses (Figure 3) and also some migration scripts had to be converted for the Derby SQL syntax (Figure 4).

[8] https://db.apache.org/derby/

```
@Override
public String getClientInformation() {
      String clientInfo = null;

      log.debug("getClientInformation: getting licenses");
      try {
            StringBuilder sqlQuery = new StringBuilder();
            String dialectName = ServerMethods.getDBRuntimeDialect().getClass().getSimpleName();
            if (dialectName.contains("Oracle")) {
                  sqlQuery.append(" select object_id as OID from user_objects where object_name=UPPER('" +
TbLicense.table_sqlName + "') ");
            } else if (dialectName.contains("PostgreSQL")) {
                  sqlQuery.append(" SELECT c.oid FROM pg_class c, pg_namespace a WHERE c.relname=LOWER('" +
TbLicense.table_sqlName + "') AND c.relnamespace = a.oid AND a.nspname=current_user");
            } else if (dialectName.contains("Derby")) {
                  sqlQuery.append(" select t.TABLEID from SYS.SYSTABLES t JOIN sys.sysschemas sc ON t.schemaid = sc.schemaid
WHERE t.TABLENAME=UPPER('" + TbLicense.table_sqlName + "') and sc.schemaname = current schema");
            } else if (dialectName.contains("SQLServer")) {
                  sqlQuery.append(" SELECT object_id  FROM sys.objects s inner join sys.schemas sh on   s.schema_id =
sh.schema_id where LOWER(s.name)=LOWER('" + TbLicense.table_sqlName + "')   and type_desc = 'USER_TABLE'  and LOWER(sh.name) =
LOWER(SCHEMA_NAME())");
            }

            Query query = getCurrentSession().createSQLQuery(sqlQuery.toString());
            clientInfo = query.uniqueResult().toString();

            log.debug("getClientInformation: get licenses successful");
      } catch (RuntimeException re) {
            log.error("getClientInformation: error getting licenses", re);
            throw re;
      }
      return clientInfo;
}
```

Figure 3 - Java method used to get the oid of the table TB_LICENSE, in order to be used to validate if the client's license is still valid

```
<sql  dbms="derby"  endDelimiter="\nGO"  splitStatements="true" stripComments="true">
      DROP SEQUENCE SEQ_DTFBILLDOCREQEXAM_ID RESTRICT;
      CREATE SEQUENCE SEQ_DTFBILLDOC_ID START WITH 3 MINVALUE 3;
</sql>

<sql  dbms="oracle"  endDelimiter="\nGO"  splitStatements="true" stripComments="true">
      DECLARE
            sequence_num NUMBER;
      BEGIN
            SELECT case when coalesce(MAX(ID), 0) + 1 >  1 then coalesce(MAX(ID), 0) + 1 else 1 end INTO sequence_num FROM
      DTF_BILL_DOC;
            EXECUTE IMMEDIATE 'DROP SEQUENCE SEQ_DTFBILLDOC_ID ';
            EXECUTE IMMEDIATE 'CREATE SEQUENCE SEQ_DTFBILLDOC_ID START WITH ' || sequence_num || ' MINVALUE ' || sequence_num
      || ' INCREMENT BY 1 NOMAXVALUE NOCYCLE NOCACHE ORDER';
      END;
</sql>
```

Figure 4 – Liquibase SQL scripts (Derby and Oracle) to recreate a sequence

The first issue of this integration was encountered during the tests with a Derby database, using the Derby's network client driver. When testing with the Derby's embedded driver[9] the application ran smoothly, without any issues. Yet, the embedded driver was not an option to use with the SafeCloud SQ1 Engine because, when an application accesses a database using the Embedded Derby JDBC driver, the Derby engine does not run in a separate process, i.e., it runs inside the same Java Virtual Machine (JVM) as the application. This aspect was problematic because we could have multiple web servers running CLINIdATA® accessing the database and with the embedded driver, this was not possible. The solution for this issue was to use the Derby Network Client[10] which provides network connectivity so multiple clients can access the same Derby database. However, Derby Network Client does not support nested savepoints yet, as it is reported in the Apache bug tracking system[11], and the application crashed with a SQLException on the first test. CLINIdATA® uses nested savepoints to avoid having to rollback an entire transaction in specific situations. For now, the solution provided by INESC TEC, which is based on Derby, was to deactivate the nested savepoints feature on SafeCloud SQ1 in order to circumvent the error and to proceed with the integration.

[9] Embedded Derby, https://db.apache.org/derby/papers/DerbyTut/embedded_intro.html

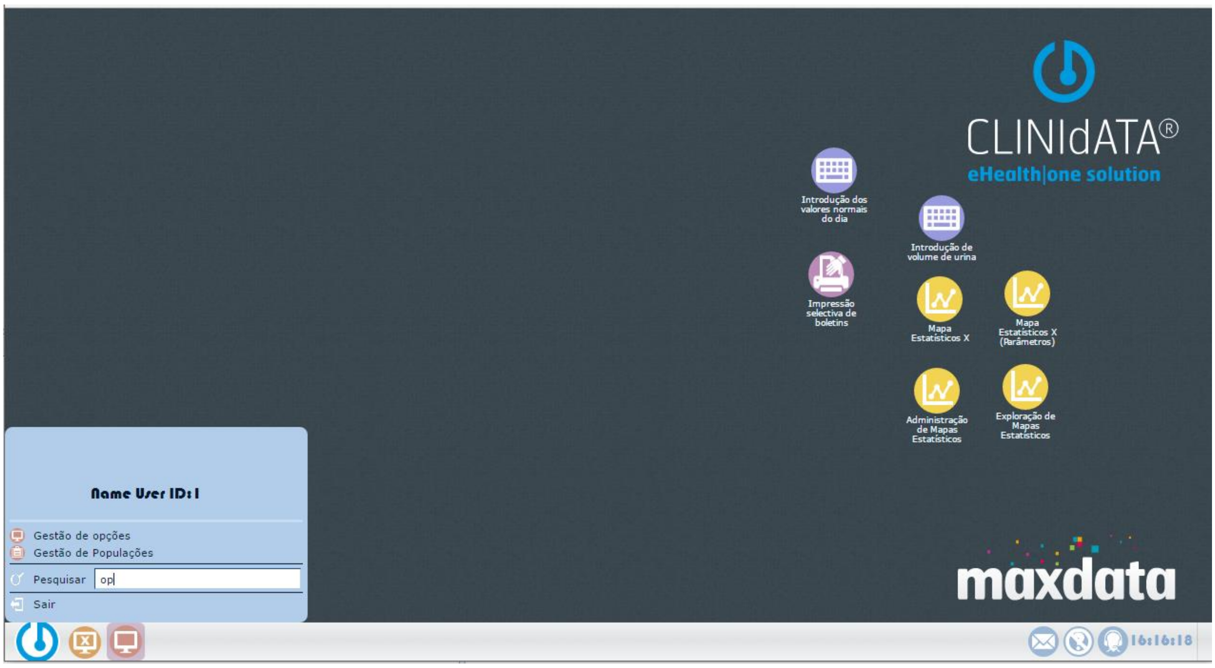
[10] Derby Network Client, https://db.apache.org/derby/papers/DerbyClientSpec.html

[11] "Make client driver allow nested savepoints", https://issues.apache.org/jira/browse/DERBY-3687

With the SafeCloud SQL Engine up and running, the first step consisted on the creation of the database required for the CLINIdATA® web application. For this purpose, we used a migration tool called Liquibase[12] that reads the schema from xml files and creates it on a given database. After the creation of the schema, Liquibase is also used to populate the schema with some sample data, read from xml files as well.

The next step of the integration, was to run the CLINIdATA® web application using the SafeCloud SQL Engine, with the database already created and populated with some data to run a few tests. Here, another issue arose when the application was trying to start. The Hibernate ORM Framework provides an automatic auditing system[13] that is created automatically on the first time the application runs and it is maintained automatically every time the domain changes. When trying to create the auditing system, some SQL statements were rejected because SafeCloud SQ1 does not support them yet, for instance statements such as "ALTER TABLE COLUMN (…)". These statements are executed automatically by Hibernate. If the auditing system is activated, when the application is running, it tries to insert data on the auditing tables as operations are done on the system and if an auditing table doesn't exist, an error occurs. To avoid this issue, we deactivated the auditing system until the problem is solved and proceeded to the next step.

Finally, with the application running, the icons were not shown on the UI, given that SafeCloud SQ1 does not support BLOBs[14] yet. CLINIdATA® stores some icons and also binary data related with the reports, on the database, as BLOBs. Comparing Figure 5 with Figure 6, it is noticeable the visual impact that the icons have on the application's UI.


CLINIdATA®
eHealth|one solution
Introdução dos valores normais do dia
Introdução de volume de urina
Impressão selectiva de boletins
Mapa Estatísticos X
Mapa Estatísticos X (Parâmetros)
Administração de Mapas Estatísticos
Exploração de Mapas Estatísticos
maxdata
Name User ID: 1
Gestão de opções
Gestão de Populações
Sair
16:16:18

Pesquisar: op

Figure 5 - CLINIdATA® main screen, using an embedded Derby database.

[12] http://www.liquibase.org/

[13] https://docs.jboss.org/envers/docs/#preface

[14] https://en.wikipedia.org/wiki/Binary_large_object

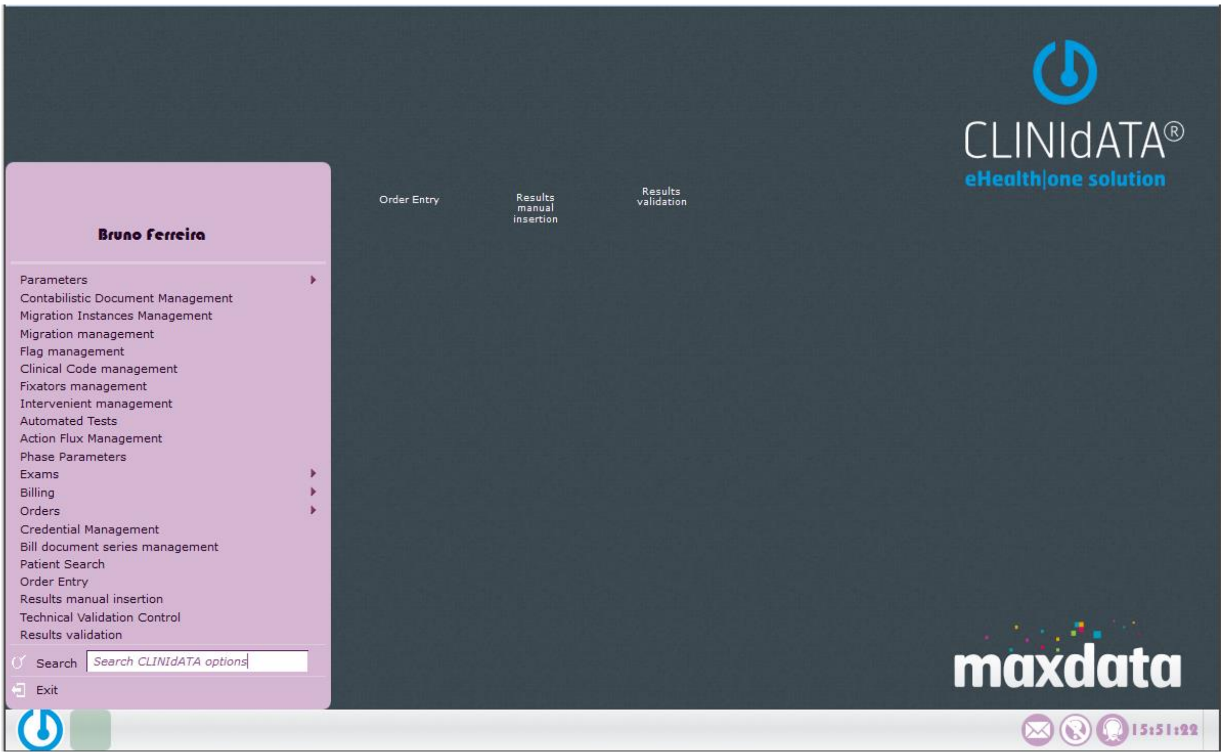
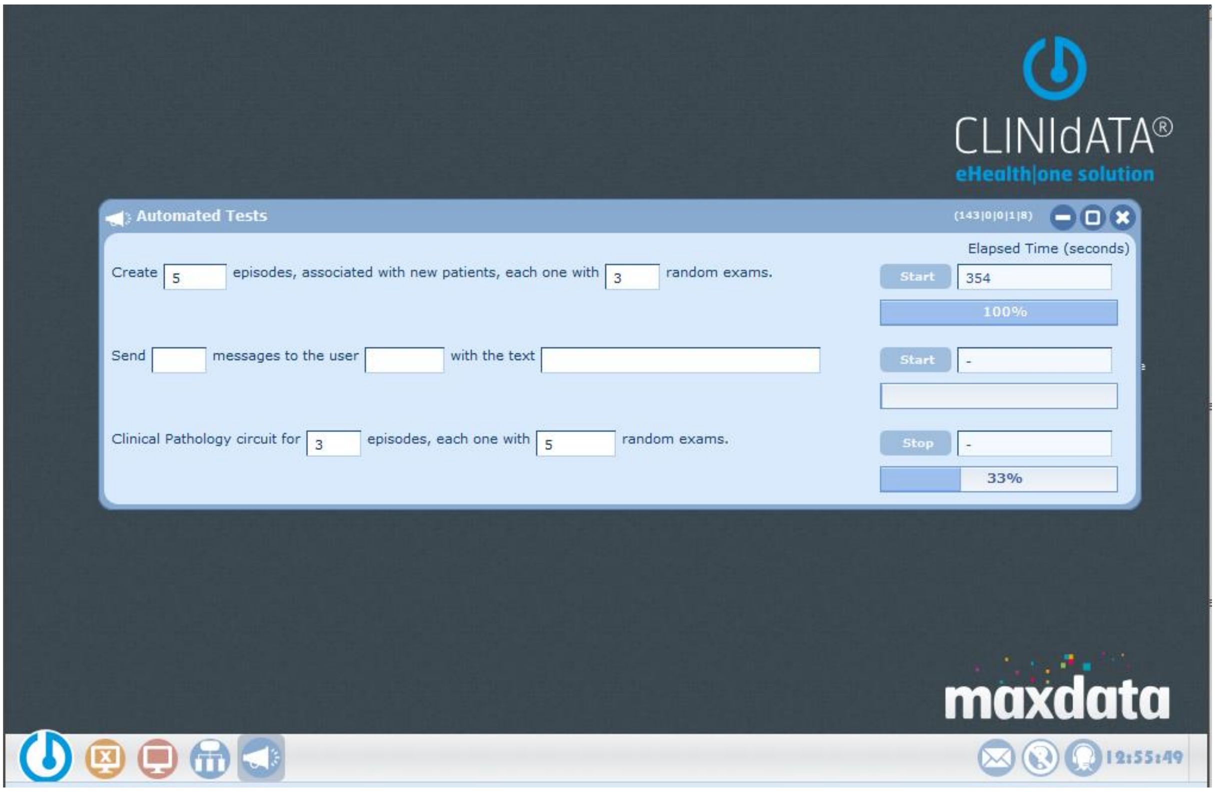


Figure 6 - CLINIdATA® main screen, integrated with SafeCloud SQ1 (BLOBs not supported).

Despite of this latter issue, the full circuit of a typical laboratory was tested successfully with the creation of multiple patient registrations with random exams, insertion and validation of the results. To easily test the full circuit multiple times, a utility option (Automated Tests - Figure 7) was developed so that the user could trigger a set of automated tests with random data and check the progress and total time of the operation.



Elapsed Time (seconds)

Create 5 episodes, associated with new patients, each one with 3 random exams. Start 354 100%

Send messages to the user with the text Start -

Clinical Pathology circuit for 3 episodes, each one with 5 random exams. Stop - 33%

Figure 7 - Utility to automatically test the full circuit of a typical laboratory.

We conducted a set of initial experiments in order to evaluate the performance overhead of the current version of SafeCloud SQ1 regarding a normal embedded Apache Derby database. SafeCloud SQ1 was tested in 2 scenarios:

- Without security features activated
- With security features activated, namely by encrypting a set of table columns related to patient identification and exam results

Regarding to the data that should be covered by the security features for this prototype, the columns listed on Table 1 were considered as crucial.

TABLE	COLUMN	DESCRIPTION
DTW_PATIENT	**NAME**	Name of the patient.
DTW_PATIENT	**CALCULATED_BIRTHDATE_STAMP**	Calculated birthdate for the patient in particular cases where the patient doesn't know the exact birthdate.
DTW_PATIENT	**BIRTHDATE_STAMP**	Patient's birthdate.
DTW_PATIENT_ID_BY_PATIENT	**SUBJECT_ID**	Set of IDs that can be used to identify a patient, e.g., social security number, citizen id.
DTW_TEST_RESULT	**NORMAL_VALUE**	Value of a specific result for an exam.

Table 1 - Columns protected by the SQ1 security features.

Three CLINIdATA® workloads were used:

- Creation and insertion of 1 episode with 1 random exam
- Transmission of 100 messages with the text "*Dear Alice, this message is a test. Thanks, Bruno.*"
- Clinical Pathology Full Circuit for 1 episode with 1 random exam

Table 2 presents the number of seconds CLINIdATA® took to complete each workload when using: Apache Derby, SafeCloud SQ1 without security features, and SafeCloud SQ1 with the security features mentioned above.

	Creation of episodes and exams (seconds)	Message transmission (seconds)	Clinical Pathology Full Circuit (seconds)
Apache Derby	3	6	6
SafeCloud SQ1 without security features	1884	7	2039
SafeCloud SQ1 with security features	3809	7	3260

Table 2 - Elapsed time and overhead of the different test scenarios.

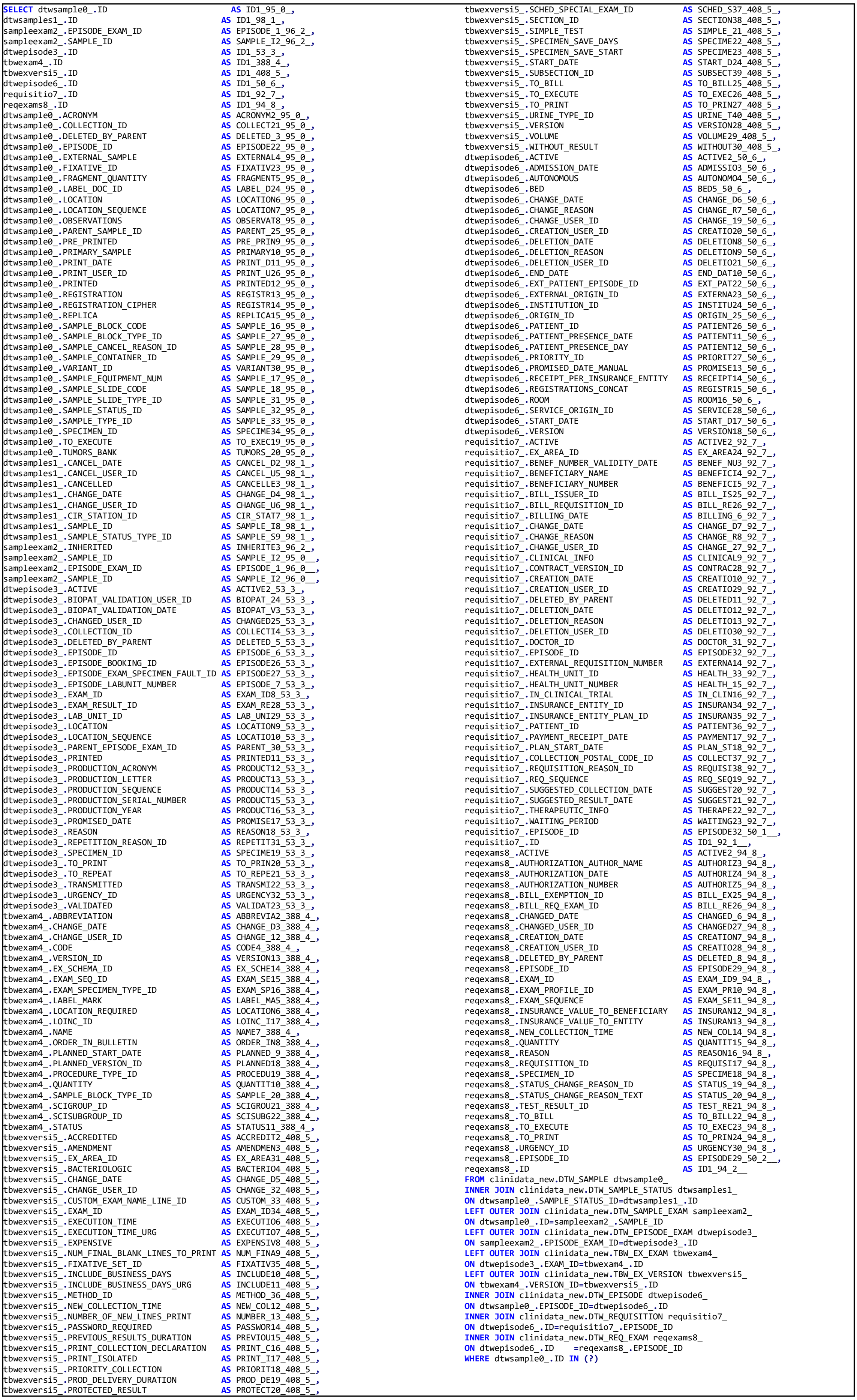
The results show that SafeCloud SQ1 without security features activated introduces a performance overhead across all the operations when compared with Apache Derby and this overhead increases significantly when the security features are activated.

The creation of episodes and exams, and the Clinical Pathology full circuit, have more complexity than the message transmission. That's why it was the operation that registered the lower discrepancy when comparing the three storage solutions.

Analysing the logs produced during the tests we verified that some queries performed poorly when executed by SafeCloud SQ1, for instance if the query returned a large number of columns. Figure 8 presents a query that took several minutes to execute either with or without the security features. This query was triggered during the execution of the test that creates a new episode with a random exam and also during the Clinical Pathology full circuit.

The described tests were executed on an environment with the following specs:

- Intel i3 CPU, 4 cores 3.7 GHz, 8 GB ram, 128 SSD
- Ubuntu 16.04
- Docker version 17.03.1-ce



```sql
SELECT dtwsample0_.ID                       AS ID1_95_0_,
dtwsamples1_.ID                             AS ID1_98_1_,
sampleexam2_.EPISODE_EXAM_ID                AS EPISODE_1_96_2_,
sampleexam2_.SAMPLE_ID                      AS SAMPLE_I2_96_2_,
dtwepisode3_.ID                             AS ID1_53_3_,
tbwexam4_.ID                                AS ID1_388_4_,
tbwexversi5_.ID                             AS ID1_408_5_,
dtwepisode6_.ID                             AS ID1_50_6_,
requisitio7_.ID                             AS ID1_92_7_,
reqexams8_.ID                               AS ID1_94_8_,
dtwsample0_.ACRONYM                         AS ACRONYM2_95_0_,
dtwsample0_.COLLECTION_ID                   AS COLLECT21_95_0_,
dtwsample0_.DELETED_BY_PARENT               AS DELETED_3_95_0_,
dtwsample0_.EPISODE_ID                      AS EPISODE22_95_0_,
dtwsample0_.EXTERNAL_SAMPLE                 AS EXTERNAL4_95_0_,
dtwsample0_.FIXATIVE_ID                     AS FIXATIV23_95_0_,
dtwsample0_.FRAGMENT_QUANTITY               AS FRAGMENT5_95_0_,
dtwsample0_.LABEL_DOC_ID                    AS LABEL_D24_95_0_,
dtwsample0_.LOCATION                        AS LOCATION6_95_0_,
dtwsample0_.LOCATION_SEQUENCE               AS LOCATION7_95_0_,
dtwsample0_.OBSERVATIONS                    AS OBSERVAT8_95_0_,
dtwsample0_.PARENT_SAMPLE_ID                AS PARENT_25_95_0_,
dtwsample0_.PRE_PRINTED                     AS PRE_PRIN9_95_0_,
dtwsample0_.PRIMARY_SAMPLE                  AS PRIMARY10_95_0_,
dtwsample0_.PRINT_DATE                      AS PRINT_D11_95_0_,
dtwsample0_.PRINT_USER_ID                   AS PRINT_U26_95_0_,
dtwsample0_.PRINTED                         AS PRINTED12_95_0_,
dtwsample0_.REGISTRATION                    AS REGISTR13_95_0_,
dtwsample0_.REGISTRATION_CIPHER             AS REGISTR14_95_0_,
dtwsample0_.REPLICA                         AS REPLICA15_95_0_,
dtwsample0_.SAMPLE_BLOCK_CODE               AS SAMPLE_16_95_0_,
dtwsample0_.SAMPLE_BLOCK_TYPE_ID            AS SAMPLE_27_95_0_,
dtwsample0_.SAMPLE_CANCEL_REASON_ID         AS SAMPLE_28_95_0_,
dtwsample0_.SAMPLE_CONTAINER_ID             AS SAMPLE_29_95_0_,
dtwsample0_.VARIANT_ID                      AS VARIANT30_95_0_,
dtwsample0_.SAMPLE_EQUIPMENT_NUM            AS SAMPLE_17_95_0_,
dtwsample0_.SAMPLE_SLIDE_CODE               AS SAMPLE_18_95_0_,
dtwsample0_.SAMPLE_SLIDE_TYPE_ID            AS SAMPLE_31_95_0_,
dtwsample0_.SAMPLE_STATUS_ID                AS SAMPLE_32_95_0_,
dtwsample0_.SAMPLE_TYPE_ID                  AS SAMPLE_33_95_0_,
dtwsample0_.SPECIMEN_ID                     AS SPECIME34_95_0_,
dtwsample0_.TO_EXECUTE                      AS TO_EXEC19_95_0_,
dtwsample0_.TUMORS_BANK                     AS TUMORS_20_95_0_,
dtwsamples1_.CANCEL_DATE                    AS CANCEL_D2_98_1_,
dtwsamples1_.CANCEL_USER_ID                 AS CANCEL_U5_98_1_,
dtwsamples1_.CANCELLED                      AS CANCELLE3_98_1_,
dtwsamples1_.CHANGE_DATE                    AS CHANGE_D4_98_1_,
dtwsamples1_.CHANGE_USER_ID                 AS CHANGE_U6_98_1_,
dtwsamples1_.CIR_STATION_ID                 AS CIR_STAT7_98_1_,
dtwsamples1_.SAMPLE_ID                      AS SAMPLE_I8_98_1_,
dtwsamples1_.SAMPLE_STATUS_TYPE_ID          AS SAMPLE_S9_98_1_,
sampleexam2_.INHERITED                      AS INHERITE3_96_2_,
sampleexam2_.SAMPLE_ID                      AS SAMPLE_I2_95_0__,
sampleexam2_.EPISODE_EXAM_ID                AS EPISODE_1_96_0__,
sampleexam2_.SAMPLE_ID                      AS SAMPLE_I2_96_0__,
dtwepisode3_.ACTIVE                         AS ACTIVE2_53_3_,
dtwepisode3_.BIOPAT_VALIDATION_USER_ID      AS BIOPAT_24_53_3_,
dtwepisode3_.BIOPAT_VALIDATION_DATE         AS BIOPAT_V3_53_3_,
dtwepisode3_.CHANGED_USER_ID                AS CHANGED25_53_3_,
dtwepisode3_.COLLECTION_ID                  AS COLLECTI4_53_3_,
dtwepisode3_.DELETED_BY_PARENT              AS DELETED_5_53_3_,
dtwepisode3_.EPISODE_ID                     AS EPISODE_6_53_3_,
dtwepisode3_.EPISODE_BOOKING_ID             AS EPISODE26_53_3_,
dtwepisode3_.EPISODE_EXAM_SPECIMEN_FAULT_ID AS EPISODE27_53_3_,
dtwepisode3_.EPISODE_LABUNIT_NUMBER         AS EPISODE_7_53_3_,
dtwepisode3_.EXAM_ID                        AS EXAM_ID8_53_3_,
dtwepisode3_.EXAM_RESULT_ID                 AS EXAM_RE28_53_3_,
dtwepisode3_.LAB_UNIT_ID                    AS LAB_UNI29_53_3_,
dtwepisode3_.LOCATION                       AS LOCATION9_53_3_,
dtwepisode3_.LOCATION_SEQUENCE              AS LOCATIO10_53_3_,
dtwepisode3_.PARENT_EPISODE_EXAM_ID         AS PARENT_30_53_3_,
dtwepisode3_.PRINTED                        AS PRINTED11_53_3_,
dtwepisode3_.PRODUCTION_ACRONYM             AS PRODUCT12_53_3_,
dtwepisode3_.PRODUCTION_LETTER              AS PRODUCT13_53_3_,
dtwepisode3_.PRODUCTION_SEQUENCE            AS PRODUCT14_53_3_,
dtwepisode3_.PRODUCTION_SERIAL_NUMBER       AS PRODUCT15_53_3_,
dtwepisode3_.PRODUCTION_YEAR                AS PRODUCT16_53_3_,
dtwepisode3_.PROMISED_DATE                  AS PROMISE17_53_3_,
dtwepisode3_.REASON                         AS REASON18_53_3_,
dtwepisode3_.REPETITION_REASON_ID           AS REPETIT31_53_3_,
dtwepisode3_.SPECIMEN_ID                    AS SPECIME19_53_3_,
dtwepisode3_.TO_PRINT                       AS TO_PRIN20_53_3_,
dtwepisode3_.TO_REPEAT                      AS TO_REPE21_53_3_,
dtwepisode3_.TRANSMITTED                    AS TRANSMI22_53_3_,
dtwepisode3_.URGENCY_ID                     AS URGENCY32_53_3_,
dtwepisode3_.VALIDATED                      AS VALIDAT23_53_3_,
tbwexam4_.ABBREVIATION                      AS ABBREVIA2_388_4_,
tbwexam4_.CHANGE_DATE                       AS CHANGE_D3_388_4_,
tbwexam4_.CHANGE_USER_ID                    AS CHANGE_12_388_4_,
tbwexam4_.CODE                              AS CODE4_388_4_,
tbwexam4_.VERSION_ID                        AS VERSION13_388_4_,
tbwexam4_.EX_SCHEMA_ID                      AS EX_SCHE14_388_4_,
tbwexam4_.EXAM_SEQ_ID                       AS EXAM_SE15_388_4_,
tbwexam4_.EXAM_SPECIMEN_TYPE_ID             AS EXAM_SP16_388_4_,
tbwexam4_.LABEL_MARK                        AS LABEL_MA5_388_4_,
tbwexam4_.LOCATION_REQUIRED                 AS LOCATION6_388_4_,
tbwexam4_.LOINC_ID                          AS LOINC_I17_388_4_,
tbwexam4_.NAME                              AS NAME7_388_4_,
tbwexam4_.ORDER_IN_BULLETIN                 AS ORDER_IN8_388_4_,
tbwexam4_.PLANNED_START_DATE                AS PLANNED_9_388_4_,
tbwexam4_.PLANNED_VERSION_ID                AS PLANNED18_388_4_,
tbwexam4_.PROCEDURE_TYPE_ID                 AS PROCEDU19_388_4_,
tbwexam4_.QUANTITY                          AS QUANTIT10_388_4_,
tbwexam4_.SAMPLE_BLOCK_TYPE_ID              AS SAMPLE_20_388_4_,
tbwexam4_.SCIGROUP_ID                       AS SCIGROU21_388_4_,
tbwexam4_.SCISUBGROUP_ID                    AS SCISUBG22_388_4_,
tbwexam4_.STATUS                            AS STATUS11_388_4_,
tbwexversi5_.ACCREDITED                     AS ACCREDIT2_408_5_,
tbwexversi5_.AMENDMENT                      AS AMENDMEN3_408_5_,
tbwexversi5_.EX_AREA_ID                     AS EX_AREA31_408_5_,
tbwexversi5_.BACTERIOLOGIC                  AS BACTERIO4_408_5_,
tbwexversi5_.CHANGE_DATE                    AS CHANGE_D5_408_5_,
tbwexversi5_.CHANGE_USER_ID                 AS CHANGE_32_408_5_,
tbwexversi5_.CUSTOM_EXAM_NAME_LINE_ID       AS CUSTOM_33_408_5_,
tbwexversi5_.EXAM_ID                        AS EXAM_ID34_408_5_,
tbwexversi5_.EXECUTION_TIME                 AS EXECUTIO6_408_5_,
tbwexversi5_.EXECUTION_TIME_URG             AS EXECUTIO7_408_5_,
tbwexversi5_.EXPENSIVE                      AS EXPENSIV8_408_5_,
tbwexversi5_.NUM_FINAL_BLANK_LINES_TO_PRINT AS NUM_FINA9_408_5_,
tbwexversi5_.FIXATIVE_SET_ID                AS FIXATIV35_408_5_,
tbwexversi5_.INCLUDE_BUSINESS_DAYS          AS INCLUDE10_408_5_,
tbwexversi5_.INCLUDE_BUSINESS_DAYS_URG      AS INCLUDE11_408_5_,
tbwexversi5_.METHOD_ID                      AS METHOD_36_408_5_,
tbwexversi5_.NEW_COLLECTION_TIME            AS NEW_COL12_408_5_,
tbwexversi5_.NUMBER_OF_NEW_LINES_PRINT      AS NUMBER_13_408_5_,
tbwexversi5_.PASSWORD_REQUIRED              AS PASSWOR14_408_5_,
tbwexversi5_.PREVIOUS_RESULTS_DURATION      AS PREVIOU15_408_5_,
tbwexversi5_.PRINT_COLLECTION_DECLARATION   AS PRINT_C16_408_5_,
tbwexversi5_.PRINT_ISOLATED                 AS PRINT_I17_408_5_,
tbwexversi5_.PRIORITY_COLLECTION            AS PRIORIT18_408_5_,
tbwexversi5_.PROD_DELIVERY_DURATION         AS PROD_DE19_408_5_,
tbwexversi5_.PROTECTED_RESULT               AS PROTECT20_408_5_,
tbwexversi5_.SCHED_SPECIAL_EXAM_ID          AS SCHED_S37_408_5_,
tbwexversi5_.SECTION_ID                     AS SECTION38_408_5_,
tbwexversi5_.SIMPLE_TEST                    AS SIMPLE_21_408_5_,
tbwexversi5_.SPECIMEN_SAVE_DAYS             AS SPECIME22_408_5_,
tbwexversi5_.SPECIMEN_SAVE_START            AS SPECIME23_408_5_,
tbwexversi5_.START_DATE                     AS START_D24_408_5_,
tbwexversi5_.SUBSECTION_ID                  AS SUBSECT39_408_5_,
tbwexversi5_.TO_BILL                        AS TO_BILL25_408_5_,
tbwexversi5_.TO_EXECUTE                     AS TO_EXEC26_408_5_,
tbwexversi5_.TO_PRINT                       AS TO_PRIN27_408_5_,
tbwexversi5_.URINE_TYPE_ID                  AS URINE_T40_408_5_,
tbwexversi5_.VERSION                        AS VERSION28_408_5_,
tbwexversi5_.VOLUME                         AS VOLUME29_408_5_,
tbwexversi5_.WITHOUT_RESULT                 AS WITHOUT30_408_5_,
dtwepisode6_.ACTIVE                         AS ACTIVE2_50_6_,
dtwepisode6_.ADMISSION_DATE                 AS ADMISSIO3_50_6_,
dtwepisode6_.AUTONOMOUS                     AS AUTONOMO4_50_6_,
dtwepisode6_.BED                            AS BED5_50_6_,
dtwepisode6_.CHANGE_DATE                    AS CHANGE_D6_50_6_,
dtwepisode6_.CHANGE_REASON                  AS CHANGE_R7_50_6_,
dtwepisode6_.CHANGE_USER_ID                 AS CHANGE_19_50_6_,
dtwepisode6_.CREATION_USER_ID               AS CREATIO20_50_6_,
dtwepisode6_.DELETION_DATE                  AS DELETION8_50_6_,
dtwepisode6_.DELETION_REASON                AS DELETION9_50_6_,
dtwepisode6_.DELETION_USER_ID               AS DELETIO21_50_6_,
dtwepisode6_.END_DATE                       AS END_DAT10_50_6_,
dtwepisode6_.EXT_PATIENT_EPISODE_ID         AS EXT_PAT22_50_6_,
dtwepisode6_.EXTERNAL_ORIGIN_ID             AS EXTERNA23_50_6_,
dtwepisode6_.INSTITUTION_ID                 AS INSTITU24_50_6_,
dtwepisode6_.ORIGIN_ID                      AS ORIGIN_25_50_6_,
dtwepisode6_.PATIENT_ID                     AS PATIENT26_50_6_,
dtwepisode6_.PATIENT_PRESENCE_DATE          AS PATIENT11_50_6_,
dtwepisode6_.PATIENT_PRESENCE_DAY           AS PATIENT12_50_6_,
dtwepisode6_.PRIORITY_ID                    AS PRIORIT27_50_6_,
dtwepisode6_.PROMISED_DATE_MANUAL           AS PROMISE13_50_6_,
dtwepisode6_.RECEIPT_PER_INSURANCE_ENTITY   AS RECEIPT14_50_6_,
dtwepisode6_.REGISTRATIONS_CONCAT           AS REGISTR15_50_6_,
dtwepisode6_.ROOM                           AS ROOM16_50_6_,
dtwepisode6_.SERVICE_ORIGIN_ID              AS SERVICE28_50_6_,
dtwepisode6_.START_DATE                     AS START_D17_50_6_,
dtwepisode6_.VERSION                        AS VERSION18_50_6_,
requisitio7_.ACTIVE                         AS ACTIVE2_92_7_,
requisitio7_.EX_AREA_ID                     AS EX_AREA24_92_7_,
requisitio7_.BENEF_NUMBER_VALIDITY_DATE     AS BENEF_NU3_92_7_,
requisitio7_.BENEFICIARY_NAME               AS BENEFICI4_92_7_,
requisitio7_.BENEFICIARY_NUMBER             AS BENEFICI5_92_7_,
requisitio7_.BILL_ISSUER_ID                 AS BILL_IS25_92_7_,
requisitio7_.BILL_REQUISITION_ID            AS BILL_RE26_92_7_,
requisitio7_.BILLING_DATE                   AS BILLING_6_92_7_,
requisitio7_.CHANGE_DATE                    AS CHANGE_D7_92_7_,
requisitio7_.CHANGE_REASON                  AS CHANGE_R8_92_7_,
requisitio7_.CHANGE_USER_ID                 AS CHANGE_27_92_7_,
requisitio7_.CLINICAL_INFO                  AS CLINICAL9_92_7_,
requisitio7_.CONTRACT_VERSION_ID            AS CONTRAC28_92_7_,
requisitio7_.CREATION_DATE                  AS CREATIO10_92_7_,
requisitio7_.CREATION_USER_ID               AS CREATIO29_92_7_,
requisitio7_.DELETED_BY_PARENT              AS DELETED11_92_7_,
requisitio7_.DELETION_DATE                  AS DELETIO12_92_7_,
requisitio7_.DELETION_REASON                AS DELETIO13_92_7_,
requisitio7_.DELETION_USER_ID               AS DELETIO30_92_7_,
requisitio7_.DOCTOR_ID                      AS DOCTOR_31_92_7_,
requisitio7_.EPISODE_ID                     AS EPISODE32_92_7_,
requisitio7_.EXTERNAL_REQUISITION_NUMBER    AS EXTERNA14_92_7_,
requisitio7_.HEALTH_UNIT_ID                 AS HEALTH_33_92_7_,
requisitio7_.HEALTH_UNIT_NUMBER             AS HEALTH_15_92_7_,
requisitio7_.IN_CLINICAL_TRIAL              AS IN_CLIN16_92_7_,
requisitio7_.INSURANCE_ENTITY_ID            AS INSURAN34_92_7_,
requisitio7_.INSURANCE_ENTITY_PLAN_ID       AS INSURAN35_92_7_,
requisitio7_.PATIENT_ID                     AS PATIENT36_92_7_,
requisitio7_.PAYMENT_RECEIPT_DATE           AS PAYMENT17_92_7_,
requisitio7_.PLAN_START_DATE                AS PLAN_ST18_92_7_,
requisitio7_.COLLECTION_POSTAL_CODE_ID      AS COLLECT37_92_7_,
requisitio7_.REQUISITION_REASON_ID          AS REQUISI38_92_7_,
requisitio7_.REQ_SEQUENCE                   AS REQ_SEQ19_92_7_,
requisitio7_.SUGGESTED_COLLECTION_DATE      AS SUGGEST20_92_7_,
requisitio7_.SUGGESTED_RESULT_DATE          AS SUGGEST21_92_7_,
requisitio7_.THERAPEUTIC_INFO               AS THERAPE22_92_7_,
requisitio7_.WAITING_PERIOD                 AS WAITING23_92_7_,
requisitio7_.EPISODE_ID                     AS EPISODE32_50_1__,
requisitio7_.ID                             AS ID1_92_1__,
reqexams8_.ACTIVE                           AS ACTIVE2_94_8_,
reqexams8_.AUTHORIZATION_AUTHOR_NAME        AS AUTHORIZ3_94_8_,
reqexams8_.AUTHORIZATION_DATE               AS AUTHORIZ4_94_8_,
reqexams8_.AUTHORIZATION_NUMBER             AS AUTHORIZ5_94_8_,
reqexams8_.BILL_EXEMPTION_ID                AS BILL_EX25_94_8_,
reqexams8_.BILL_REQ_EXAM_ID                 AS BILL_RE26_94_8_,
reqexams8_.CHANGED_DATE                     AS CHANGED_6_94_8_,
reqexams8_.CHANGED_USER_ID                  AS CHANGED27_94_8_,
reqexams8_.CREATION_DATE                    AS CREATION7_94_8_,
reqexams8_.CREATION_USER_ID                 AS CREATIO28_94_8_,
reqexams8_.DELETED_BY_PARENT                AS DELETED_8_94_8_,
reqexams8_.EPISODE_ID                       AS EPISODE29_94_8_,
reqexams8_.EXAM_ID                          AS EXAM_ID9_94_8_,
reqexams8_.EXAM_PROFILE_ID                  AS EXAM_PR10_94_8_,
reqexams8_.EXAM_SEQUENCE                    AS EXAM_SE11_94_8_,
reqexams8_.INSURANCE_VALUE_TO_BENEFICIARY   AS INSURAN12_94_8_,
reqexams8_.INSURANCE_VALUE_TO_ENTITY        AS INSURAN13_94_8_,
reqexams8_.NEW_COLLECTION_TIME              AS NEW_COL14_94_8_,
reqexams8_.QUANTITY                         AS QUANTIT15_94_8_,
reqexams8_.REASON                           AS REASON16_94_8_,
reqexams8_.REQUISITION_ID                   AS REQUISI17_94_8_,
reqexams8_.SPECIMEN_ID                      AS SPECIME18_94_8_,
reqexams8_.STATUS_CHANGE_REASON_ID          AS STATUS_19_94_8_,
reqexams8_.STATUS_CHANGE_REASON_TEXT        AS STATUS_20_94_8_,
reqexams8_.TEST_RESULT_ID                   AS TEST_RE21_94_8_,
reqexams8_.TO_BILL                          AS TO_BILL22_94_8_,
reqexams8_.TO_EXECUTE                       AS TO_EXEC23_94_8_,
reqexams8_.TO_PRINT                         AS TO_PRIN24_94_8_,
reqexams8_.URGENCY_ID                       AS URGENCY30_94_8_,
reqexams8_.EPISODE_ID                       AS EPISODE29_50_2__,
reqexams8_.ID                               AS ID1_94_2__
FROM clinidata_new.DTW_SAMPLE dtwsample0_
INNER JOIN clinidata_new.DTW_SAMPLE_STATUS dtwsamples1_
ON dtwsample0_.SAMPLE_STATUS_ID=dtwsamples1_.ID
LEFT OUTER JOIN clinidata_new.DTW_SAMPLE_EXAM sampleexam2_
ON dtwsample0_.ID=sampleexam2_.SAMPLE_ID
LEFT OUTER JOIN clinidata_new.DTW_EPISODE_EXAM dtwepisode3_
ON sampleexam2_.EPISODE_EXAM_ID=dtwepisode3_.ID
LEFT OUTER JOIN clinidata_new.TBW_EX_EXAM tbwexam4_
ON dtwepisode3_.EXAM_ID=tbwexam4_.ID
LEFT OUTER JOIN clinidata_new.TBW_EX_VERSION tbwexversi5_
ON tbwexam4_.VERSION_ID=tbwexversi5_.ID
INNER JOIN clinidata_new.DTW_EPISODE dtwepisode6_
ON dtwsample0_.EPISODE_ID=dtwepisode6_.ID
INNER JOIN clinidata_new.DTW_REQUISITION requisitio7_
ON dtwepisode6_.ID=requisitio7_.EPISODE_ID
INNER JOIN clinidata_new.DTW_REQ_EXAM reqexams8_
ON dtwepisode6_.ID    =reqexams8_.EPISODE_ID
WHERE dtwsample0_.ID IN (?)
```

Figure 8 - Query that performed slowly during the creation of a new episode.

3.2.2 Secure SQL Engine Integration Work

As explained above, some features required by CLINIdATA® are not currently supported by SafeCloud's Secure Database Server. The most prominent of the missing features are *savepoints* and BLOBs. Whenever these features are required, they are currently being bypassed as CLINIdATA® is able to work without them. In fact, as observable in the previous sections, it is already possible to test the application on top of the platform and identify the need for a thorough performance evaluation and optimization. The work on optimization will be a priority in the next project period. After the platform reaches the expected level of performance, integration work will focus on full support of the missing features.

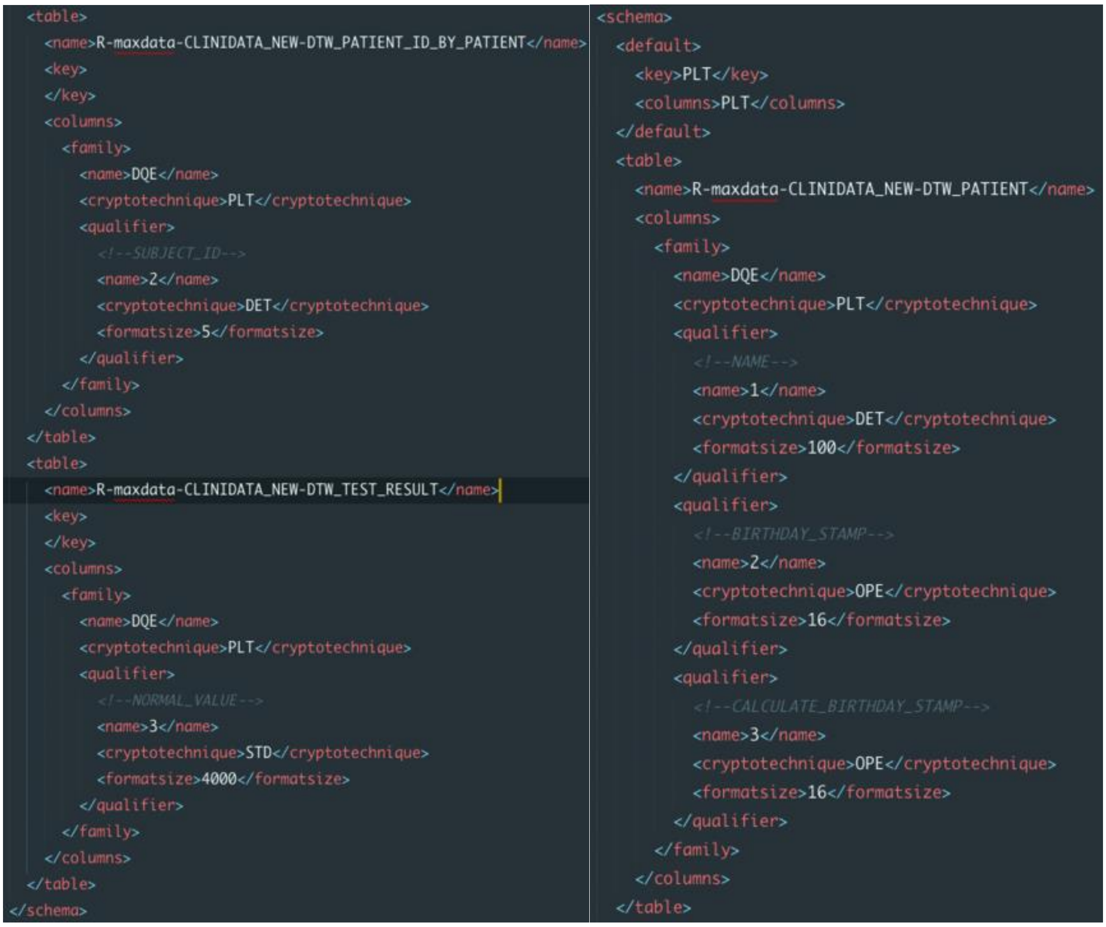
Besides the need to bypass some currently incomplete features, the bulk of integration effort put into running CLINIdATA® on top of the SafeCloud platform is related with configuration and parametrization. The Secure Database Server is designed to work as an *application-agnostic* system aiming at supporting the majority of SQL-based workloads. However, in order to provide adequate privacy-preserving security levels it is necessary to identify, for a particular data schema, which data is sensitive and which data is not. This way, it is possible to instantiate the database with a set of privacy-preserving techniques that offer the ideal compromise between privacy and performance. As a consequence, in order to integrate CLINIdATA® with SafeCloud's Secure Database Server, Maxdata had to provide a list of the columns that were considered to contain sensitive data. With such list (depicted in Table 1), INESC TEC produced a configuration file that describes how CLINIdATA®'s database is going to be protected. This configuration file is depicted in Figure 9. As observable, the configuration file does not contain an explicit reference to the tables and columns defined previously. This is related to the fact that, at the trusted deployment, the SafeCloud SQL Engine translates SQL requests to secure NoSQL operations that will be performed at the untrusted deployment (third-party cloud infrastructure). This process also involves translating the SQL schema, including the name of the tables (Table 3) and columns (Table 4), to NoSQL ones.

SQL Table	NoSQL Table
DTW_PATIENT	R-maxdata-CLINIDATA_NEW-DTW_PATIENT
DTW_PATIENT_ID_BY_PATIENT	R-maxdata-CLINIDATA_NEW-DTW_PATIENT_ID_BY_PATIENT
DTW_TEST_RESULT	R-maxdata-CLINIDATA_NEW-DTW_TEST_RESULT

Table 3 - SQL to NoSQL table translation.

SQL Columns	NoSQL Columns (Column Family – Qualifier)
NAME	DQE-1
BIRTHDAY_STAMP	DQE-2
CALCULATE_BIRTHDAY_STAMP	DQE-3
SUBJECT_ID	DQE-2
NORMAL_VALUE	DQE-3

Table 4 - SQL to NoSQL Columns translation.



```xml
  <table>
    <name>R-maxdata-CLINIDATA_NEW-DTW_PATIENT_ID_BY_PATIENT</name>
    <key>
    </key>
    <columns>
      <family>
        <name>DQE</name>
        <cryptotechnique>PLT</cryptotechnique>
        <qualifier>
          <!--SUBJECT_ID-->
          <name>2</name>
          <cryptotechnique>DET</cryptotechnique>
          <formatsize>5</formatsize>
        </qualifier>
      </family>
    </columns>
  </table>
  <table>
    <name>R-maxdata-CLINIDATA_NEW-DTW_TEST_RESULT</name>
    <key>
    </key>
    <columns>
      <family>
        <name>DQE</name>
        <cryptotechnique>PLT</cryptotechnique>
        <qualifier>
          <!--NORMAL_VALUE-->
          <name>3</name>
          <cryptotechnique>STD</cryptotechnique>
          <formatsize>4000</formatsize>
        </qualifier>
      </family>
    </columns>
  </table>
</schema>
```

```xml
<schema>
  <default>
    <key>PLT</key>
    <columns>PLT</columns>
  </default>
  <table>
    <name>R-maxdata-CLINIDATA_NEW-DTW_PATIENT</name>
    <columns>
      <family>
        <name>DQE</name>
        <cryptotechnique>PLT</cryptotechnique>
        <qualifier>
          <!--NAME-->
          <name>1</name>
          <cryptotechnique>DET</cryptotechnique>
          <formatsize>100</formatsize>
        </qualifier>
        <qualifier>
          <!--BIRTHDAY_STAMP-->
          <name>2</name>
          <cryptotechnique>OPE</cryptotechnique>
          <formatsize>16</formatsize>
        </qualifier>
        <qualifier>
          <!--CALCULATE_BIRTHDAY_STAMP-->
          <name>3</name>
          <cryptotechnique>OPE</cryptotechnique>
          <formatsize>16</formatsize>
        </qualifier>
      </family>
    </columns>
  </table>
```

Figure 9 – SafeCloud Configuration File.

The translated names and qualifiers are then used to compile the configuration file. Such file is used by the database to ensure that queries on sensitive data are always translated into queries that preserve such data private even when data is stored and processed at the untrusted deployment (third-party cloud infrastructure). The system was also designed to allow the definition of a privacy level by default in order to avoid specifying a privacy-preserving technique per table when, potentially, there are thousands of them. This way it is possible to focus the configuration work on data that is the most sensitive and important to protect.

4 Demonstration Procedure

This section presents a set of demonstration procedures that show in practice how the integration described in the previous section were used to implement a set of reference use cases described in Deliverable D5.2.

4.1 Insert exam order

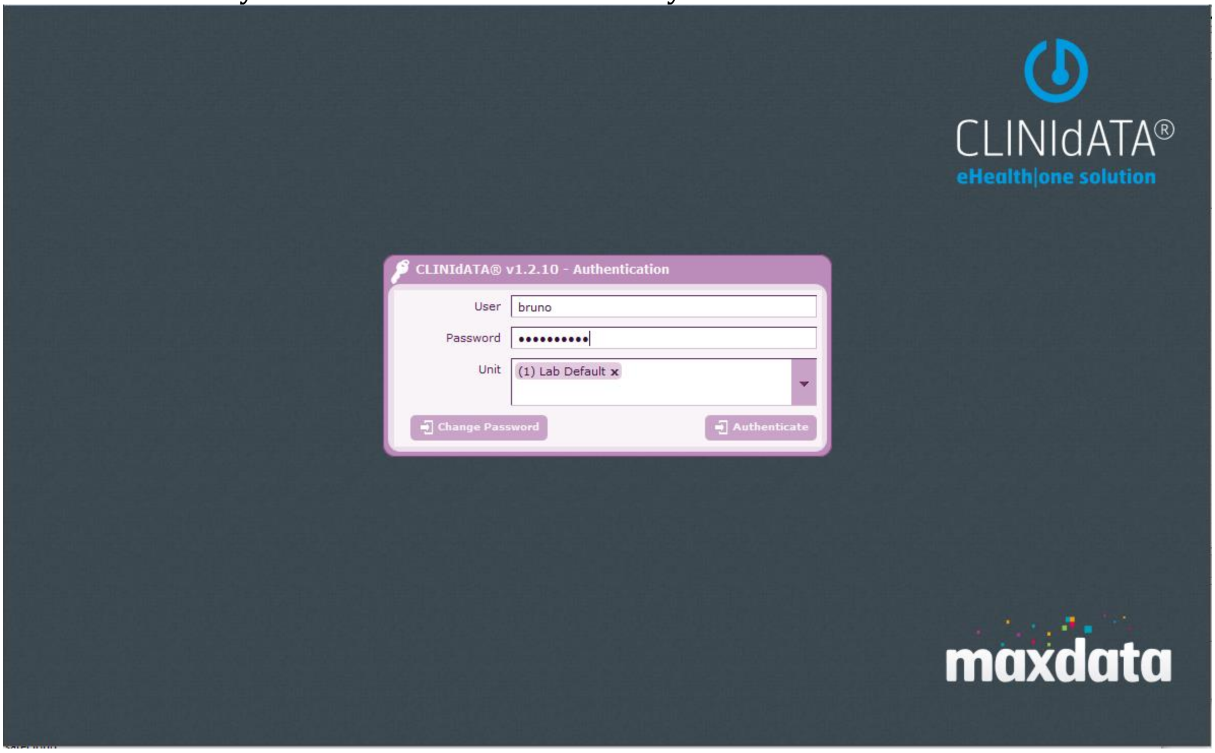
This demonstration procedure is related to the use-case 3.2.3.3 described in the Deliverable D5.1.

After the insertion of the user credentials and a successful login on the application (Figure 10), the main screen appears to the user. As we can see on Figure 11, it looks like a classic desktop environment, with a start menu, some shortcuts, a taskbar, clock, message system and help system.

At a typical laboratory, the circuit starts with the insertion of an exam order, usually done by an administrative at the reception that searches for a given patient when he arrives at the lab.

The first step is to search for the user in the system, for instance "John Doe" (Figure 12) and then select it on the next screen that shows the search results (Figure 13). On Figure 14 we can see some data about the patient already filled on the screen of the option used to register new patients on the system. This information was gathered on a previous visit and it is used by default for the creation of a new order.

The user can then proceed to the insertion of an exam, selecting it on the text field on the bottom of the screen and after the insertion it appears on the grid that is on the right side of the screen (Figure 15). Finally, the user clicks on the button "Save" to insert the new order and gets a message saying that the order was successfully saved (Figure 16). Notice that the saved order has now the id 71 (marked with a red box on Figure 16). This id will be used to identify the order inside the laboratory.



CLINIdATA® v1.2.10 - Authentication

User bruno

Password ••••••••••

Unit (1) Lab Default ×

Figure 10 - CLINIdATA® Login Screen.

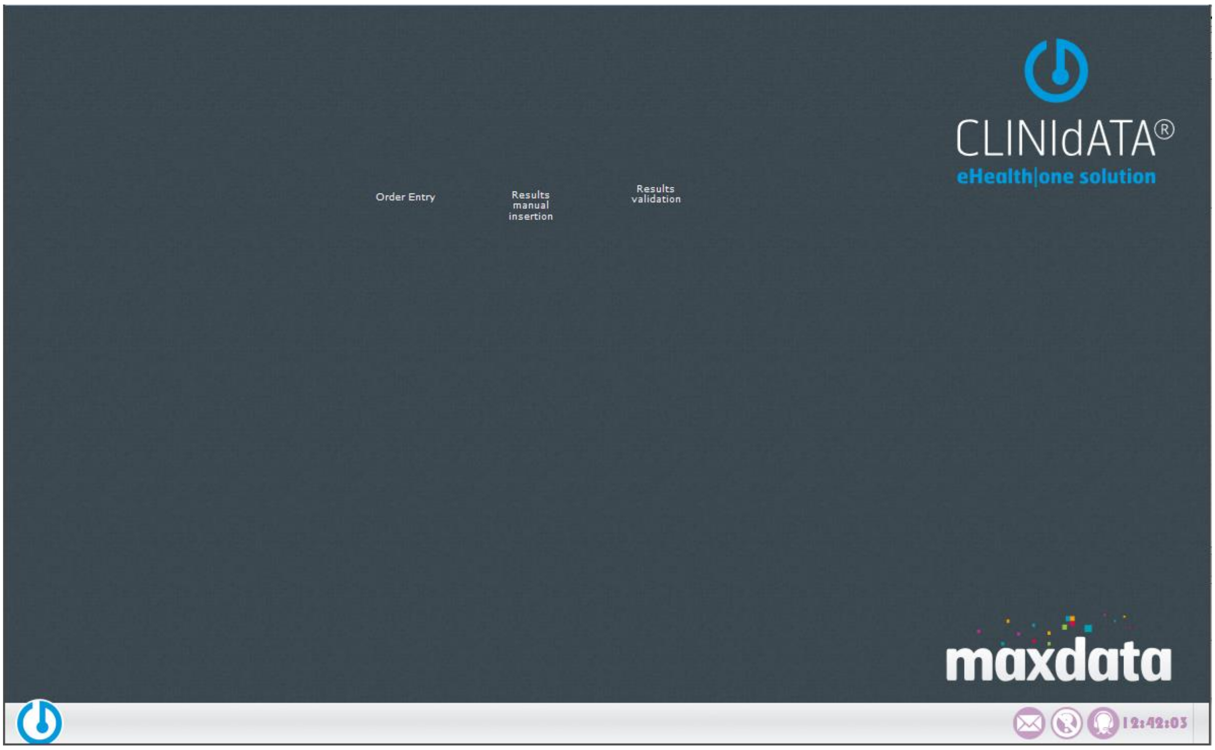
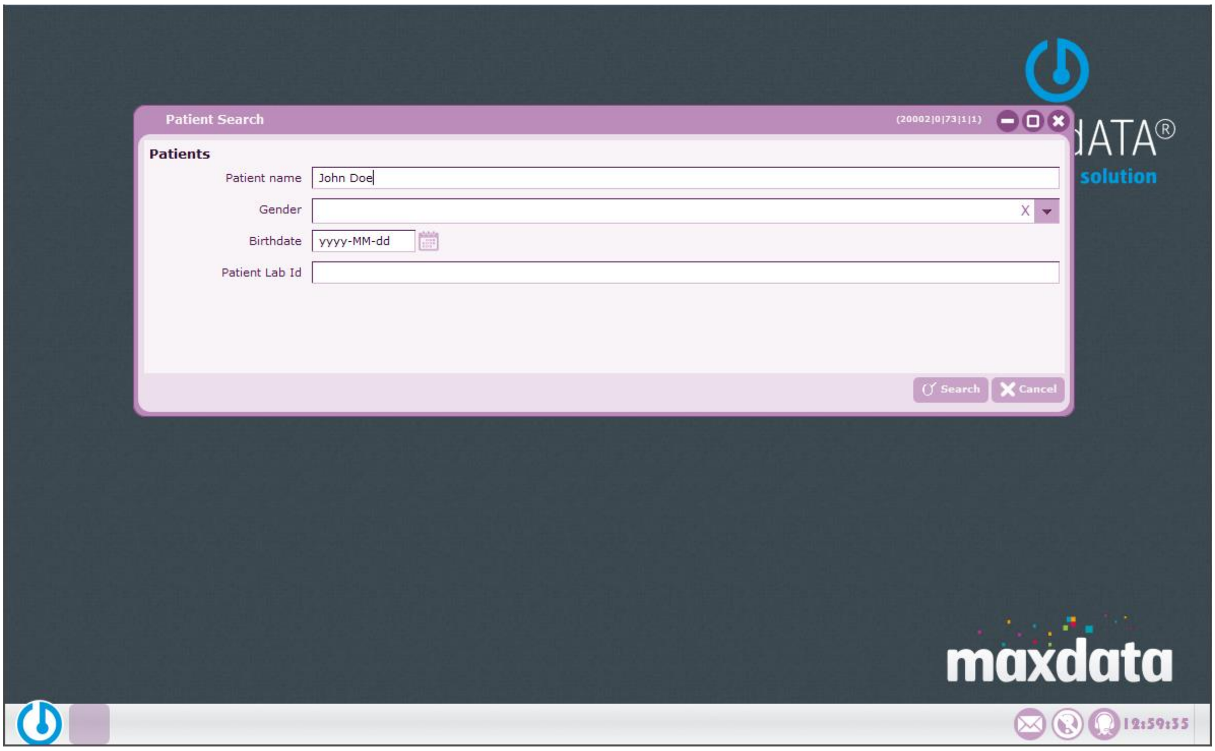


Figure 11 - CLINIdATA® main screen (no icons shown due to missing BLOB support by SafeCloud SQ1).


Patient Search

Patients

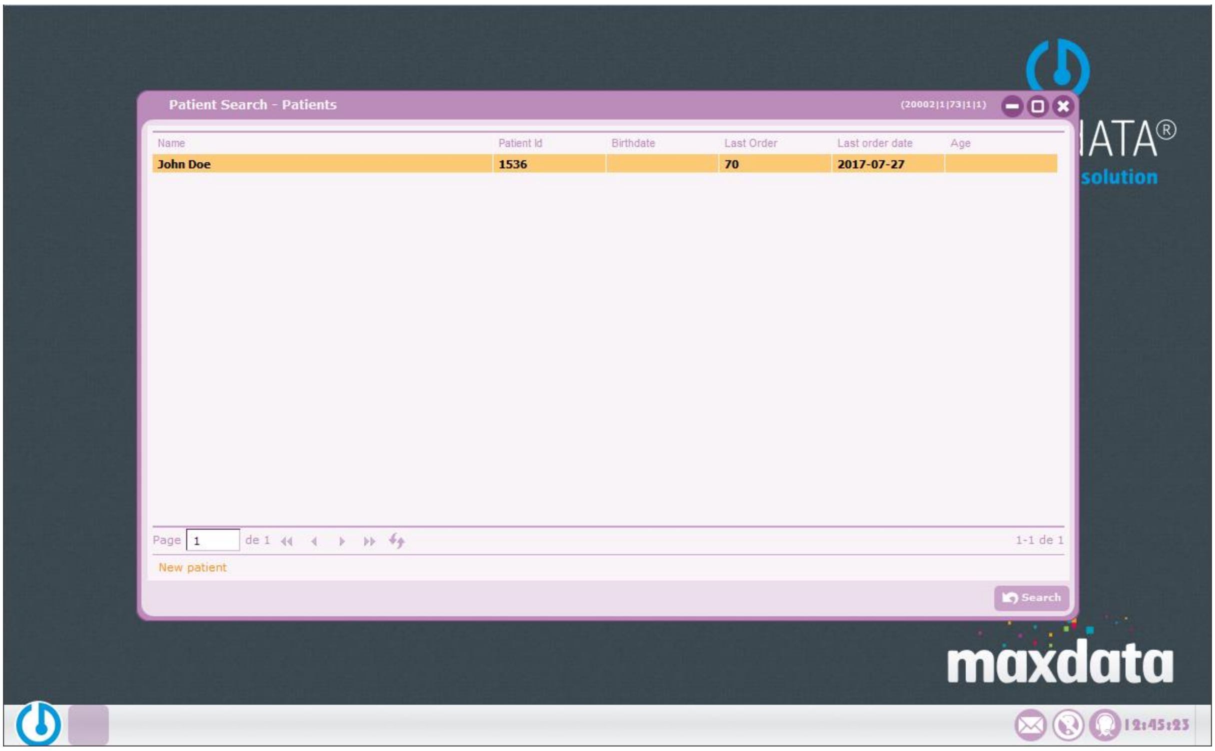
Patient name: John Doe

Gender:

Birthdate: yyyy-MM-dd

Patient Lab Id:

Figure 12 - Option used to search for a patient.



Patient Search - Patients

(20002|1|73|1|1)

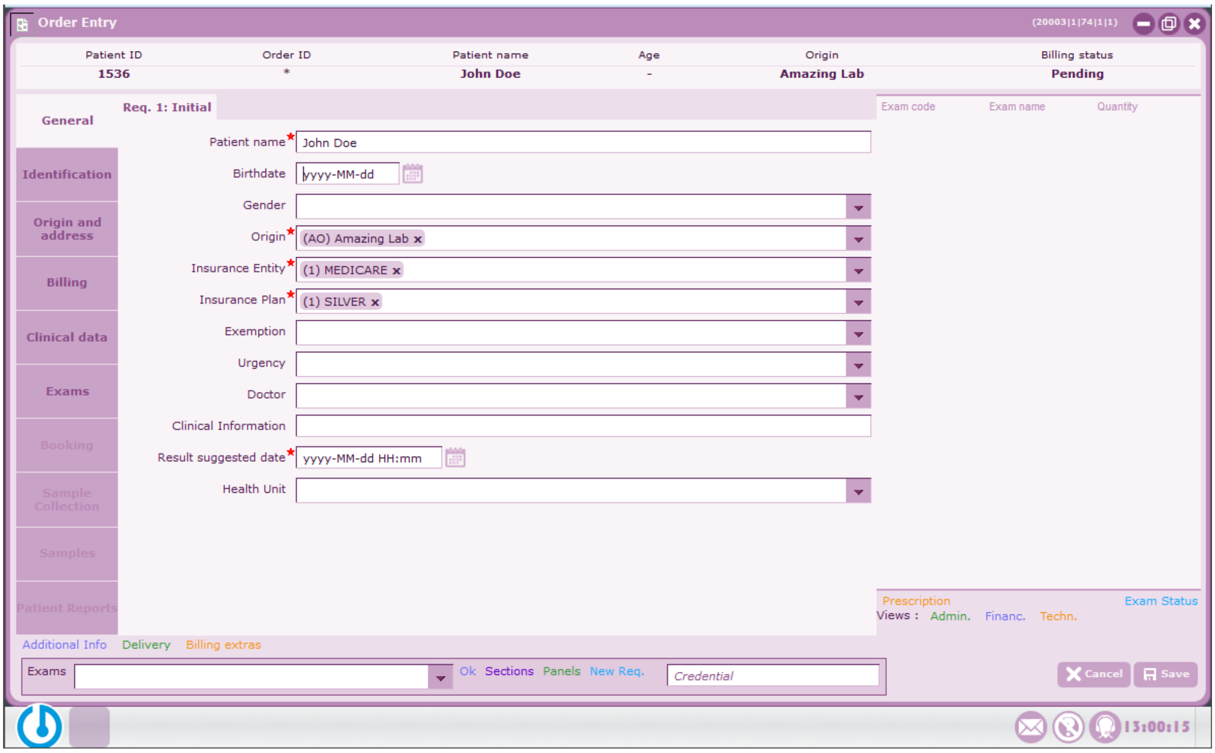
Name	Patient Id	Birthdate	Last Order	Last order date	Age
John Doe	1536		70	2017-07-27	

Page 1 de 1

1-1 de 1

New patient

Figure 13 - Search results for the patient “John Doe”.



Order Entry

(20003|1|74|1|1)

Patient ID	Order ID	Patient name	Age	Origin	Billing status
1536	*	John Doe	-	Amazing Lab	Pending

Req. 1: Initial

Patient name* John Doe

Birthdate yyyy-MM-dd

Gender

Origin* (AO) Amazing Lab ×

Insurance Entity* (1) MEDICARE ×

Insurance Plan* (1) SILVER ×

Exemption

Urgency

Doctor

Clinical Information

Result suggested date* yyyy-MM-dd HH:mm

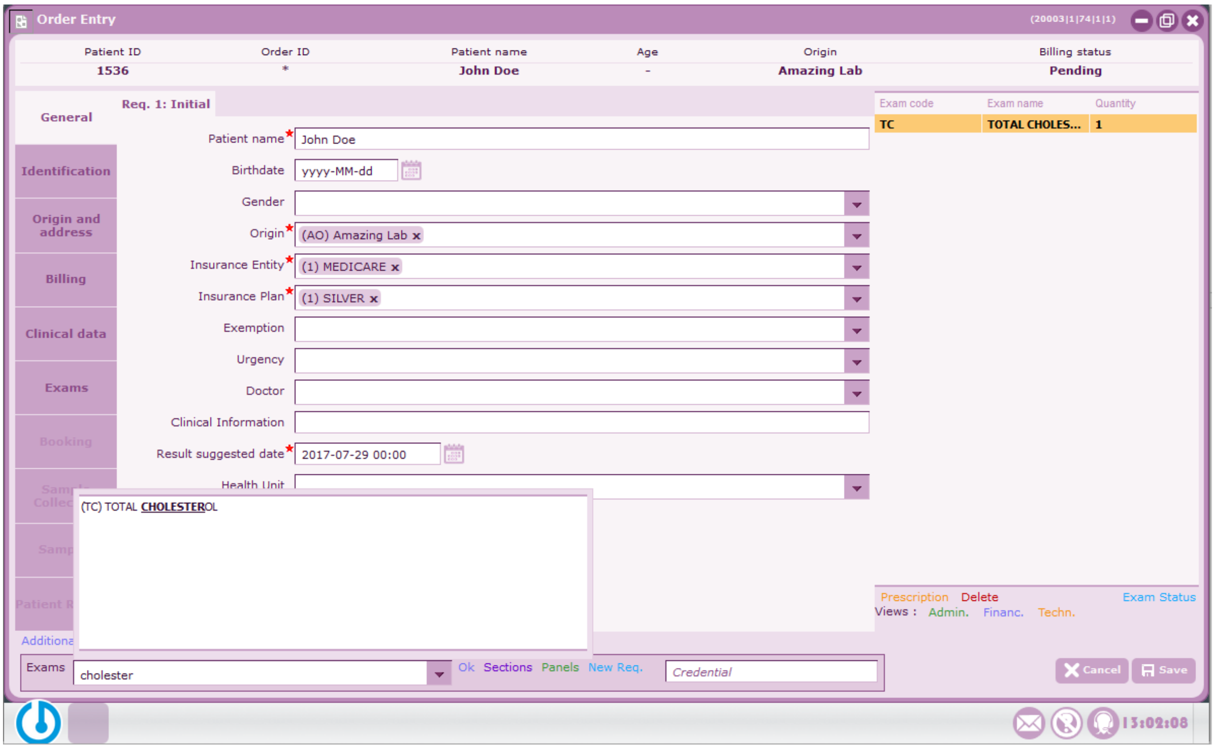
Health Unit

Exam code	Exam name	Quantity

Exams

Credential

Figure 14 – “Order Entry” creation screen.



Patient ID	Order ID	Patient name	Age	Origin	Billing status
1536	*	John Doe	-	Amazing Lab	Pending

Patient name* John Doe

Birthdate yyyy-MM-dd

Gender

Origin* (AO) Amazing Lab

Insurance Entity* (1) MEDICARE

Insurance Plan* (1) SILVER

Exemption

Urgency

Doctor

Clinical Information

Result suggested date* 2017-07-29 00:00

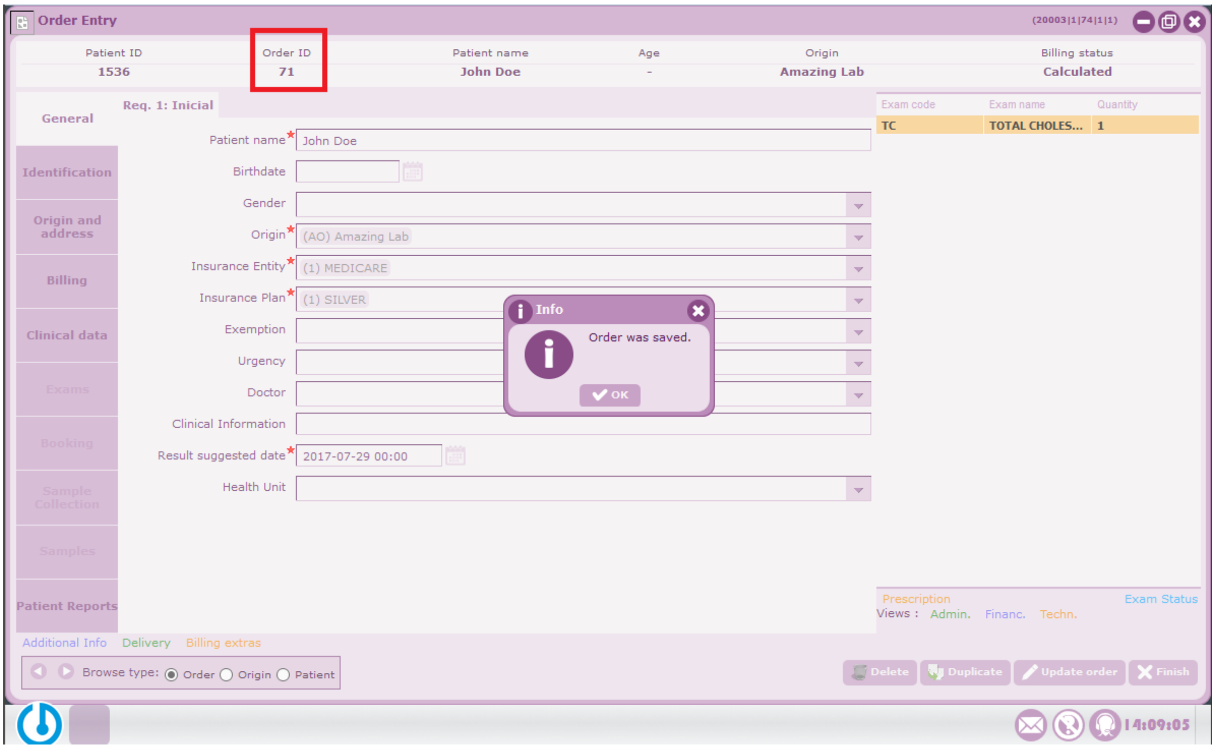
Health Unit

(TC) TOTAL CHOLESTEROL

Exams cholester

Exam code	Exam name	Quantity
TC	TOTAL CHOLES...	1

Figure 15 - Exam insertion on the "Order Entry" option.



Patient ID	Order ID	Patient name	Age	Origin	Billing status
1536	71	John Doe	-	Amazing Lab	Calculated

Patient name* John Doe

Birthdate

Gender

Origin* (AO) Amazing Lab

Insurance Entity* (1) MEDICARE

Insurance Plan* (1) SILVER

Exemption

Urgency

Doctor

Clinical Information

Result suggested date* 2017-07-29 00:00

Health Unit

Exam code	Exam name	Quantity
TC	TOTAL CHOLES...	1

Info

Order was saved.

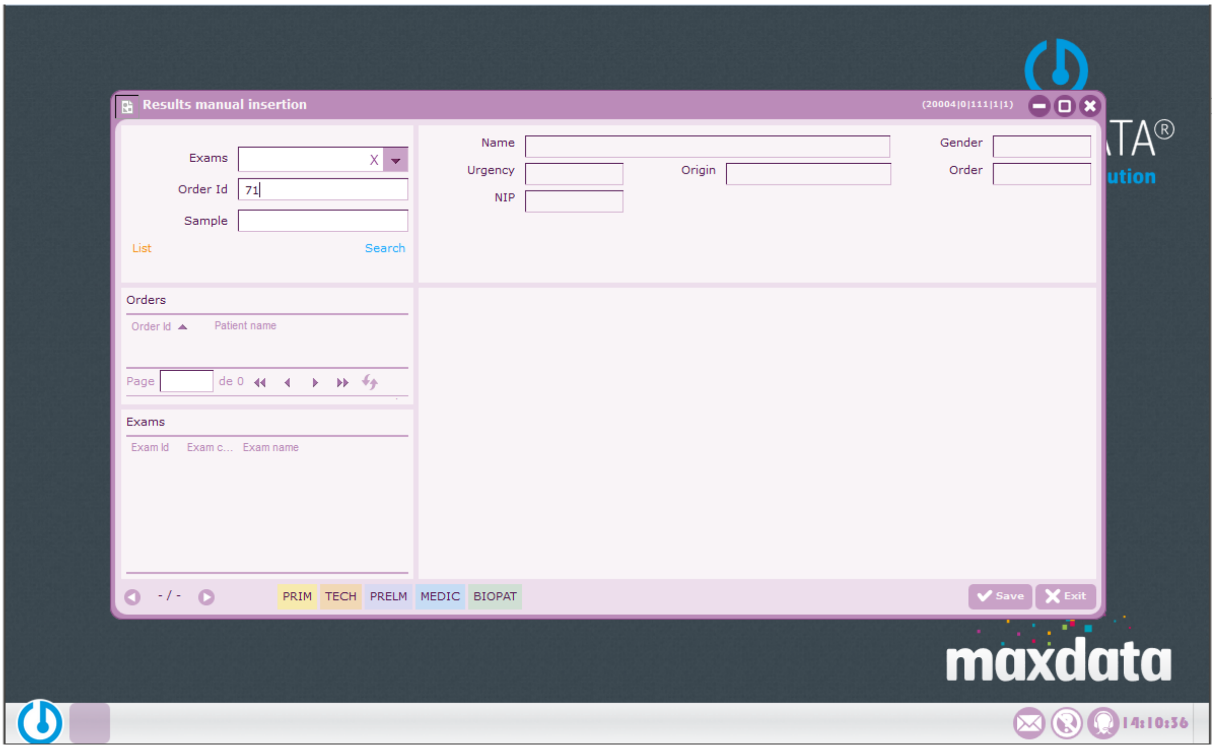
Browse type: ● Order ○ Origin ○ Patient

Figure 16 - Order successfully saved.

4.2 Insert exam result

This demonstration procedure is related to the use-case 3.2.3.3 described in the Deliverable D5.2.

After the creation of the exam order, the doctor at the laboratory can proceed with the insertion of the exam results on the system, whenever they’re ready. To do that, the doctor uses the “Results manual insertion” option to search for the order with the id 71 (Figure 17). After the search, the screen refreshes with some information about the patient and with new fields to insert the result values. In this case, everything's ok with the patient and the doctor appoints it on the "Exam comments" field, after filling the result for the total cholesterol that was 80 mg/dl (Figure 18). Finally, the user clicks on the "Save button" and the title of the exam is marked with the orange colour, that matches the colour used for the technical validation as it is visible on the bottom of the screen (Figure 19).



Results manual insertion (20004|0|111|1|1)

Exams: X
Order Id: 71
Sample:

Name:
Urgency:
Origin:
NIP:
Gender:
Order:

Orders

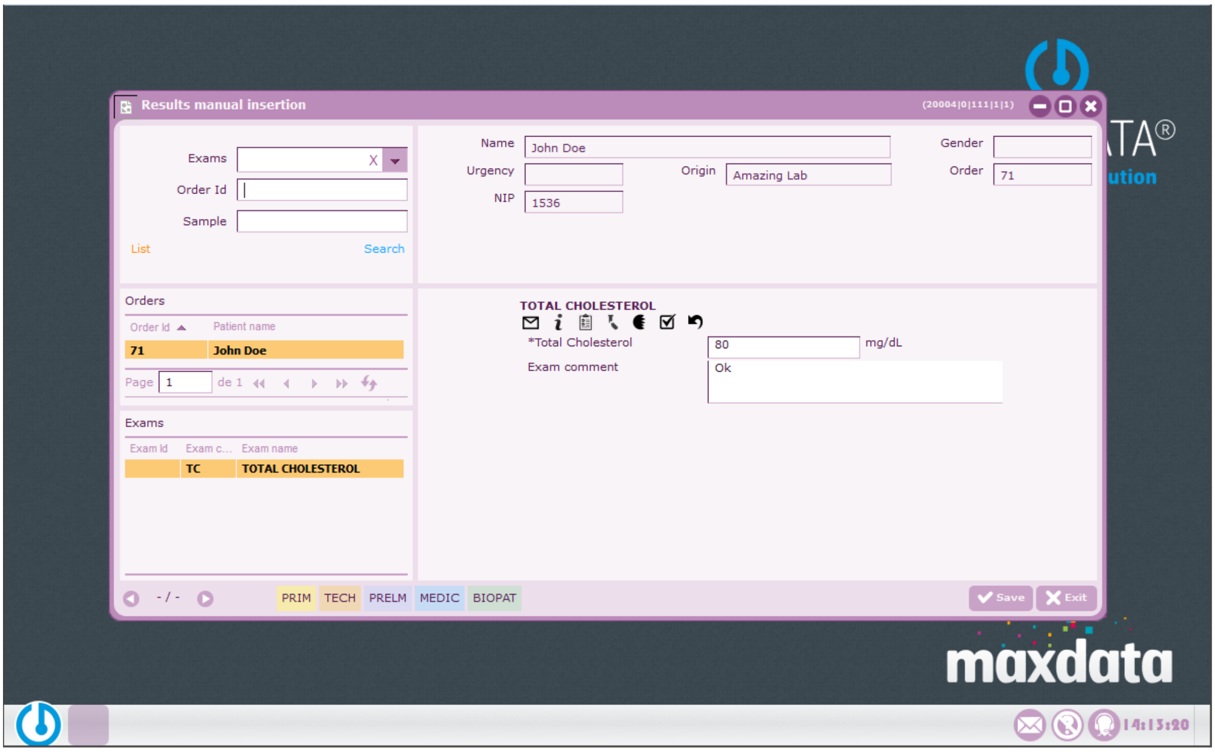
Order Id	Patient name

Page de 0

Exams

Exam Id	Exam c...	Exam name

Figure 17 - "Results insertion" option.



Results manual insertion (20004|0|111|1|1)

Exams: X
Order Id:
Sample:

Name: John Doe
Urgency:
Origin: Amazing Lab
NIP: 1536
Gender:
Order: 71

Orders

Order Id	Patient name
71	John Doe

Page 1 de 1

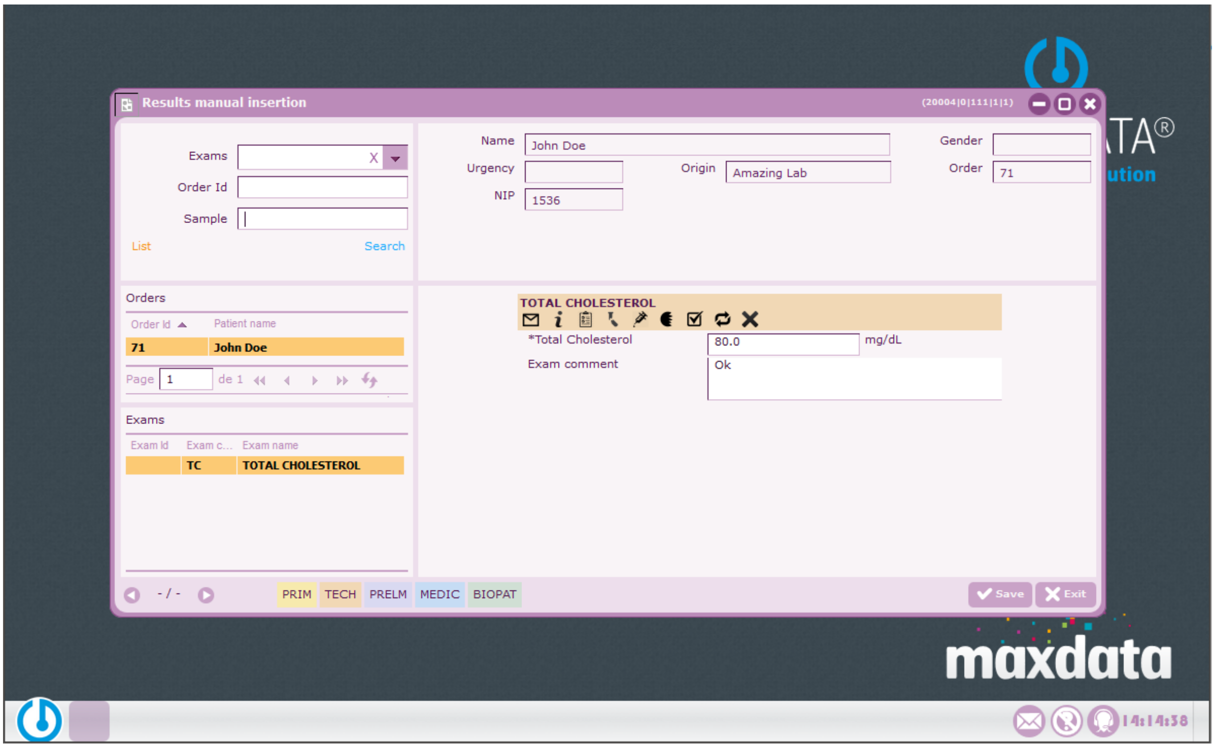
Exams

Exam Id	Exam c...	Exam name
	TC	TOTAL CHOLESTEROL

TOTAL CHOLESTEROL

*Total Cholesterol: 80 mg/dL
Exam comment: Ok

Figure 18 - Result insertion for the exam "Total Cholesterol".



Results manual insertion (20004|0|111|1|1)

Name: John Doe
Urgency:
Origin: Amazing Lab
NIP: 1536
Gender:
Order: 71

Orders

Order Id	Patient name
71	John Doe

Page 1 de 1

Exams

Exam Id	Exam c...	Exam name
	TC	TOTAL CHOLESTEROL

TOTAL CHOLESTEROL

*Total Cholesterol: 80.0 mg/dL
Exam comment: Ok

Figure 19 - Result saved for the exam "Total Cholesterol".

4.3 Access exam result

This demonstration procedure is related to the use-case 3.2.3.3 described in the Deliverable D5.2.

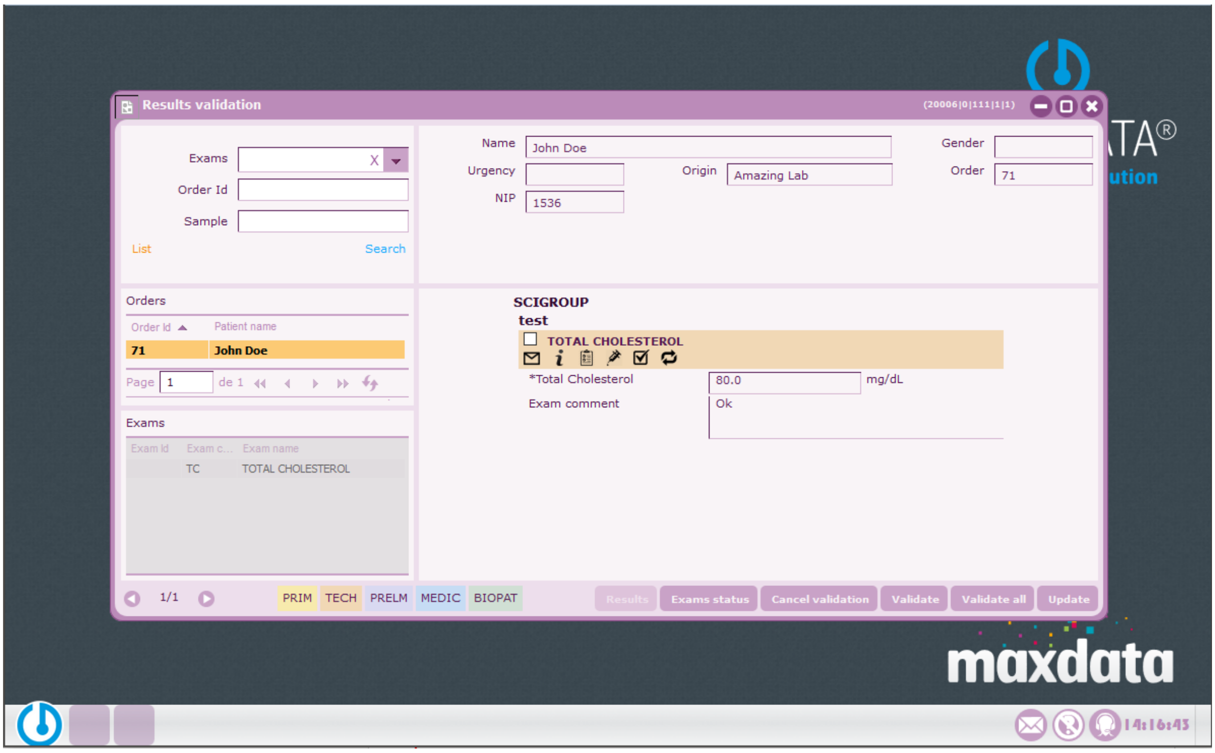
With the result inserted, another doctor on the laboratory could then also validate the result so it could have a higher level of validation, for instance medical or biopathological validation (Figure 20). To validate all the exams the user could click on the button “Validate all” on the bottom of the screen and the name of the exam (“Total Cholesterol”) would now be highlighted with the color blue (Figure 21).

On this option the user can also send the exam for repetition, for instance if he finds something wrong on the values.

4.4 Access production control

This demonstration procedure is related to the use-case 3.2.3.3 described in the Deliverable D5.2.

In order to get an overview of the number of exams already with a specific type of validation, it is possible to configure an option on CLINIdATA® to get this type of data. For instance, using the production control option the doctor could get all the orders registered on the current day, with results also inserted in the current day and with primary or technical validation (Figure 22). Clicking then on the “Search” button it is visible on the results that the order 71 was returned (Figure 23) because it matches with the previous search parameters. The doctor could then click on the “Details” that is on the “Exams” column in order to check the exams that were inserted on the corresponding order (Figure 24).


Results validation

Exams: X
Order Id:
Sample:

Name: John Doe
Urgency:
Origin: Amazing Lab
NIP: 1536
Gender:
Order: 71

Orders

Order Id	Patient name
71	John Doe

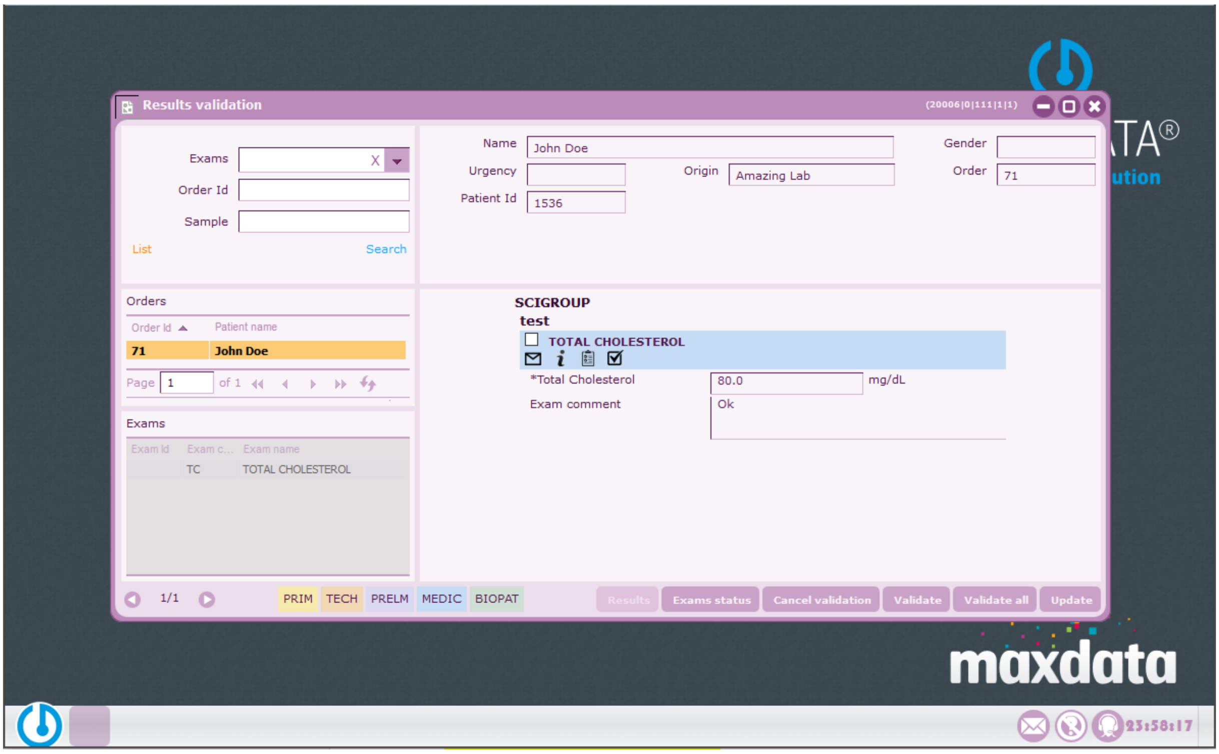
Page 1 de 1

Exams

Exam Id	Exam c...	Exam name
	TC	TOTAL CHOLESTEROL

SCIGROUP
test
□ TOTAL CHOLESTEROL
*Total Cholesterol: 80.0 mg/dL
Exam comment: Ok

Figure 20 - "Results validation” option.


Results validation

Exams: X
Order Id:
Sample:

Name: John Doe
Urgency:
Origin: Amazing Lab
Patient Id: 1536
Gender:
Order: 71

Orders

Order Id	Patient name
71	John Doe

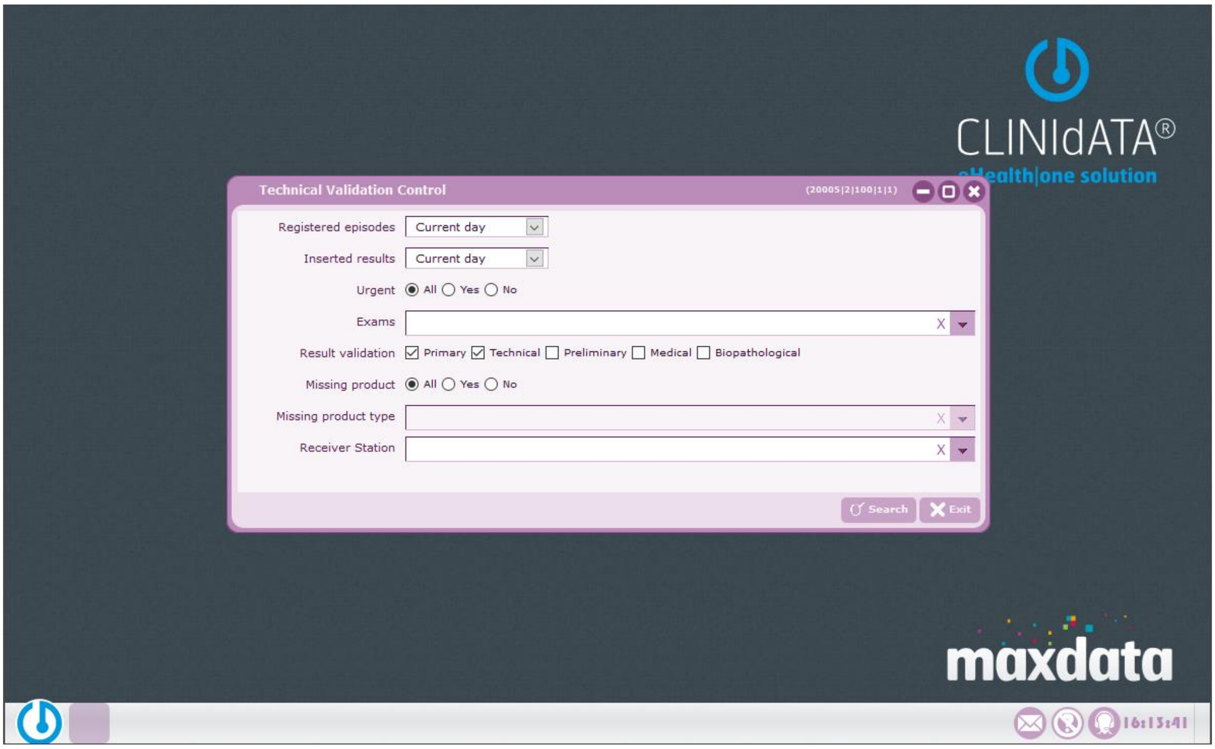
Page 1 of 1

Exams

Exam Id	Exam c...	Exam name
	TC	TOTAL CHOLESTEROL

SCIGROUP
test
□ TOTAL CHOLESTEROL
*Total Cholesterol: 80.0 mg/dL
Exam comment: Ok

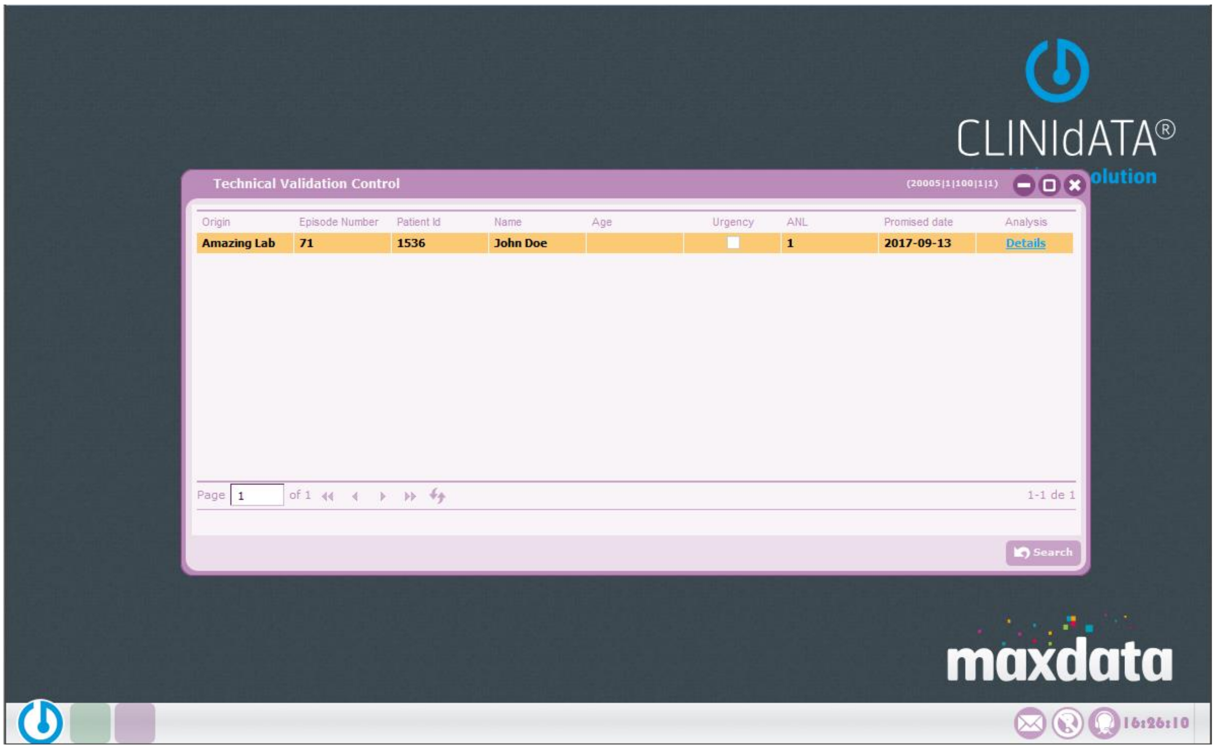
Figure 21 - Result with medical validation.



Technical Validation Control (20005|2|100|1|1)

Registered episodes	Current day
Inserted results	Current day
Urgent	● All ○ Yes ○ No
Exams	
Result validation	☑ Primary ☑ Technical □ Preliminary □ Medical □ Biopathological
Missing product	● All ○ Yes ○ No
Missing product type	
Receiver Station	

Figure 22 - "Technical Validation Control" option.

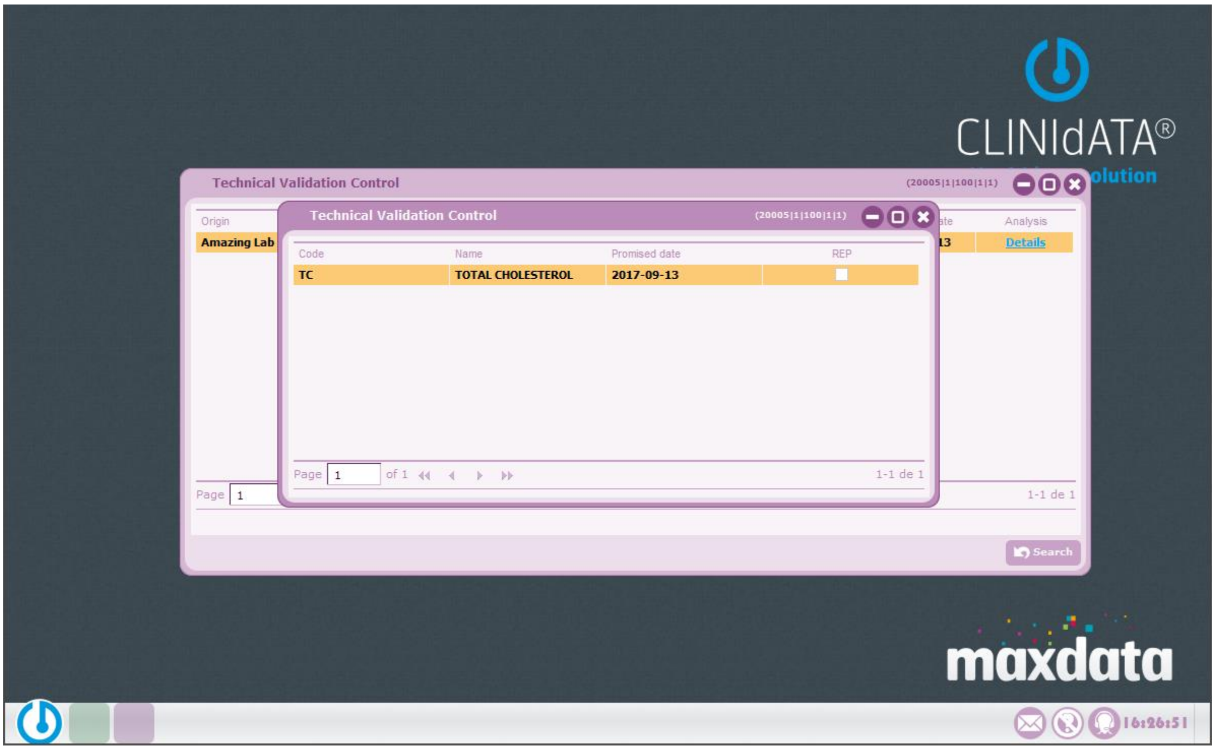


Technical Validation Control (20005|1|100|1|1)

Origin	Episode Number	Patient Id	Name	Age	Urgency	ANL	Promised date	Analysis
Amazing Lab	71	1536	John Doe		□	1	2017-09-13	Details

Page 1 of 1 1-1 de 1

Figure 23 - "Technical Validation control" search results.



Technical Validation Control (20005|1|100|1|1)

Origin	ate	Analysis
Amazing Lab	13	Details

Page 1 1-1 de 1

Technical Validation Control (20005|1|100|1|1)

Code	Name	Promised date	REP
TC	TOTAL CHOLESTEROL	2017-09-13	□

Page 1 of 1 1-1 de 1

Figure 24 - Details about the exams inserted on the order 71.

5 Conclusion

This document described a first prototype of the Maxdata SafeCloud-based healthcare platform.

The deliverable presented:

- An overview of the CLINIdATA® eHealth Solution.
- A description of the first version of the prototype and the integration done between CLINIdATA® and the Secure SQL Engine. This first version of the prototype covers 2 of the 3 scenarios presented in D5.2: the SaaS deployment and the hybrid deployment.
- A set of demonstration procedures that show in practice the implementation of a use-case described in Deliverable D5.2.

The next step will be to implement and demonstrate the remaining use cases, and improve the performance of the operations using the Secure SQL Engine. The final version of the prototype will be described in D5.6 (month 36).

6 References

[GDPR16] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) http://data.europa.eu/eli/reg/2016/679/oj