REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : 04/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Génie logiciel et Systèmes Distribués(GLSD)**

Analyse de la spécification formelle d'un processus métier

Par :

MELHAGUEG AMINA

Soutenu le ../../…. devant le jury composé de :

OUAAR Hanane	MCB	Examinateur

Année universitaire 2020-2021

بسم الله الرحمن الرحيم

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

CHAPTER 2

CHAPTER 3

TABLE OF FIGURES

CHAPTER 1

CHAPTER 2

CHAPTER 3

Thanks and Dedication

I *would like to express these few lines of thanks at the end of this humility work, and great gratitude to: We begin by thanking Mrs.* ***OUAAR Hanane****, who gave me the honor to be My coach for his follow up and his great support and all the help she gave me. I also express my sincere thanks to the* ***members of the jury*** *for their kindness examine this work and judge it.*

And I will not let this opportunity pass without my thanks the department professors, the first of whom is ***Jafal Abdel Hamid****.*

Finally, I would like to thank you all for ***helping*** *me, either directly or indirectly.*

So that my work is equal.

A special dedication to my deceased uncle, ***Shahid Al-Wajib (R. Haddouche).***

I also dedicate this humble work with love, sincerity and pride: To my ***dear family****, big and small, and first one, my* ***sister****.*

To the girlfriend who always encouraged me.

Friends: ***N. Manal, M. Younes Ilyas****.*

Without exception, for ***everything*** *we love and forget.*

Amina

ABSTRACT

The current work is based on business processes formally verification and validation. Our purpose is the design, the specification and the implementation of a simulation application for assembly-line cars. The case study has been chosen because it provides different views for several business processes: the view of the administrator that allows: system configuration, account management and components setting. The view of the technical staff that publishing the current tacked measurements, edit rapports to understand the situation of the current system, locate the damage and to on-site maintenance.

The analysis phase identifies the system business processes components (robotic arms and supply chain). The design phase use UML diagram (Diagram Class, use case diagram, sequence diagram, and activity diagram). The specification phase adopts UPPAL software tool to elaborate timed automaton in order to: verify formally the system comportment by CTL, system synchronization, business processes simulation and validation the most proprieties. The implementation phase uses the most appropriate software tools for developing the simulation system.

Keywords: Business processes, assembly-line cars, robotic arms, supply chain application, UML, UPPAL, CTL, simulation.

RESUME

Le travail actuel est fondé sur des processus opérationnels officiellement vérifiés et validés. Notre objectif est la conception, la spécification et la mise en œuvre d'une application de simulation pour les voitures de chaîne de montage. L'étude de cas a été choisie parce qu'elle fournit des vues différentes pour plusieurs processus opérationnels : la vue de l'administrateur qui permet : la configuration du système, la gestion des comptes et le réglage des composants. Le point de vue du personnel technique sur la publication des mesures actuelles, les rapports de contrôle pour comprendre la situation du système actuel, localiser les dommages et l'entretien sur place.

La phase d'analyse identifie les composants des processus opérationnels du système (bras robotisés et chaîne d'approvisionnement). La phase de conception utilise le diagramme UML (classe de diagramme, utiliser le diagramme de cas, le diagramme de séquence et le diagramme d'activité). La phase de spécification adopte l'outil logiciel UPPAL pour élaborer l'automate chronométré afin de : vérifier formellement le comportement du système par CTL, la synchronisation du système, la simulation des processus d'affaires et la validation de la plupart des proprieties. La phase de mise en œuvre utilise les outils logiciels les plus appropriés pour développer le système de simulation.

Mots clés : processus métier, voitures de la chaîne de montage, bras robotisés, application de la chaîne d'approvisionnement, administrateur, UML, UPPAL, CTL, simulation.

الـمـلـخـص

يتم تنفيذ نظام العمل الحالي على أساس العمليات التجارية. الهدف من هذا العمل هو تصميم وإنتاج خوارزميات وإحصائيات تطبيقية لسيارات خط التجميع. تم اختيار دراسة الحالة لنا. يتطلب هذا التطبيق وجهات نظر مختلفة و لكل شركة متطلباتها: تسمح طريقة عرض المسؤول بتكوين النظام وإدارة الحسابات ، يرجع الحق للموظفين الفنيين في نشر البيانات والتعليق على هذه البيانات لفهم نظام التحقق وموقع الضرر.

تحديد مرحلة التحليل لمكونات النظام (الذراع الآلية وسلسلة التوريد). ثم تطوير هذا النظام من خلال مخطط UML (مخطط الفصل ، استخدم مخطط الحالة ، مخطط التسلسل ومخطط النشاط). استخدمUPPAL لرسم التشغيل الآلي المحدد بوقت والتحقق من استخدام مرحلة تنفيذ النظام والأخذ في الاعتبار أن الرسومات قد تم تمريرها للتحقق من المحاكاة و من صحتها عن طريق (CTL) و استخدم أدوات البرامج الأكثر شيوعًا المناسبة للتطوير.

الكلمات المفتاحية: العملية التجارية ، سيارات خط التجميع ، الأسلحة الآلية ، تطبيق سلسلة التوريد ، المسؤول ، المحاكاة .UML ، UPPAL ،CTL.

GENERAL INTRODUCTION

GENERAL INTRODUCTION

Information System (IS) [34], is an integrated set of components to collect, store and process data and to provide information, knowledge and digital products. Business enterprises and other organizations depend on information systems to conduct and manage their operations, interact with their customers and suppliers, and compete in the marketplace. Information systems are used to run cross-organizational supply chains and electronic marketplaces. For example, corporations use information systems to process financial accounts, manage their human resources, and reach potential customers with online promotions. A Business Process (BP) [35] is a series of steps performed by a group of stakeholders to achieve a concrete goal. Each step in a business process denotes a task that is assigned to a participant. It is the fundamental building block for several related ideas such as business process management, process automation. About him in recent decades, these systems have been constantly updated. Newer systems have appeared that they are able to efficiently manage their processes faster and more accurately compared to previous iterations.

Graphical user interfaces (GUI) have become the norm and replaced old text terminals lacking data. Thanks to changing approaches to design, the development of these new systems is faster than ever. When it comes to the world of modern computing, it is clear that technology is changing rapidly and links are emerging more strongly between the use of certain key technologies and business improvements process services, when we look at things before and compare them with current ones, it becomes clear that most of the companies that were equipped with these technologies and share their working through them have been more successful than other companies that did not.

Today, almost all companies are integrating widespread computer systems by owning both a web application and a mobile application, which help them, share their business Process not only with computer or laptop users but with any device possible from mobile phones. To tablets and even watches to anyone, anywhere, businesses feel stronger now than ever due to this availability. That is why the work presented aims to study and implement a generalized information system based on business processes.

For this, the problem of our work focuses on responding to a set of important questions: first, how to design a generalized a critical system based on business processes management? Second, how to create links between system’s components of their business processes for best synchronization? How to verify and to validate their properties? What are the best paradigms to be adopted? How are the right tools that will be used in the design, specification and in the implementation phases, to give the good results?

We chose an assembly line as a study case of a modernized information system because: Its characteristics are the most suitable for the aspect of our modeling. Also, it offers an application of the synchronal simulation and also because of the dynamic of their business process and also for their several properties.

The objective of our work followed an approach that will focus on: In the analysis phase, we identify the public and private business process in the assembly line of cars and describe its components. In the design phase, we focus on graphically representing business process management easily and simply using UML models. The objective is then: 1) to select an automobile assembly line as a case study, 2) to identify and design its business processes, using various types of diagrams (static behavior and dynamic behavior). In the specification phase, we verify and valid our system by a temporal state machine automat using UPPAAL as the software tool. Finally, the implementation phase, we present our application as simulation system of their business process, offered a soft handling services of all users view by different stage of setting and reporting.

This thesis is broken down as follows: In the first chapter, we have presented multiple concepts that are related to our work which are: first, the concept of business process in which we present it by defining it and mentioning its types and characteristics and some of its advantages and planning and technologies, also the ways of modeling it, the second business process management in which we also define it together with the specification of its life cycle and advantages, the third concept is the business process modeling and we present all the type that we can use in a specification and verification of form to applications.

In the second chapter, we present our case study, which is an automobile assembly line. First, we focus on the design and the modeling side of their business processes using UML describing all interne and extern interactions that exist between the different actors of our system (technician and administrator) with static views: class diagram, system functionality: use case diagram and then with dynamic views: Sequence diagrams and activity diagrams.

Second, we declare system’s variables and actions by UPPAAL, a formal model specification is made by the automaton time of their business process of assembly line cars such as: robotic arms and supply chain. After that, we verify properties of this automat with a temporal logic by the model checking using Computation Tree Logic (CTL). The validation of this automat is done by the simulation as a result of synchronized behavior. In fin we make analysis comparison between some modeling paradigms

In the third and final chapter, we specify the development tools, the languages and the programming environments that are used to make our simulation application. A study between passages for our specification towards our implementation is present in the end of this chapter.

This thesis ends with a general conclusion that summarizes our study; it also talks about the potential features that we plan to add it in a future work.

CHAPITRE 01 : STATE OF ART

CHAPTER 1: STATE OF ART

1-INTRODUCTION

For the sake of the smooth running of the company and at a time of intense competition and market fluctuations, and for the competing company to stand out. As technology becomes more competitive, the company uses many processes. Business process occupations are a group of several related activities. Others to achieve a goal, generally in an organizational context that establishes Roles and Relationships. Coordination of all the resources involved during a process in order to achieve a commercial objective, and therefore an important issue, thanks to BPM (Business Process Management). Business process management is an approach that itself encompasses a management process that supports the design, implementation, and monitoring of the processes that run the organization. Mastery of business processes is essential in corporate governance is obtained through mastery of business process management, and is a fundamental principle of traditional business management [9] [10].

We start this chapter with the concept of a business process by defining and explaining some of its basic properties, and definitions of the basics of pervasive computing systems.

2-BUSINESS PROCESS

This section provides the essential notations and definitions of the business process.

2.1- Business Process definitions

It is very common, in industrial environments that unregulated works and activities carried out are frequent and she has many people involved. In these cases, it is very useful to determine a standard procedure that everyone can follow. A business process is the definition of such a "standard" Action."

There are several definitions of the business process. The most influential are as follows:

Definition 1:

A business process is a set of linked tasks whose ultimate goal is to provide services or products to customers. Business process is also defined as a set of activities and tasks, once completed, will achieve organizational goals [1].

Definition 2:

Formally a business process: P is defined as a set of activities $VP = \{V1, \ldots , Vn\}$, a directed graph $GP = (VP , EP)$, an output function $oP : VP \rightarrow Nk$ and $\forall(u, v) \in EP$ a Boolean function $f(u,v) = Nk \rightarrow \{0, 1\}$. In this case, the process is constructed in the following way: for every completed activity u, the value $oP (u)$ is calculated and then, for every other activity v, if $f (u,v) (oP(u))$ is true, v can be executed. Of course, such a definition of business process is hard to be handled by business people but is useful for formal modeling purposes [9].

Definition 3:

A business process is a series of steps performed by a group of stakeholders to achieve a concrete goal. Each step in a business process denotes a task that is assigned to a participant. It is the fundamental building block for several related ideas such as business process management, process automation [35].

2.2-Business process planning:

The business process is designed in three levels [32]:

- Level 1: The business level shows the high-level business view of the process, with identification the main stages and their impact on the organization of the company. This level is determined by Business-style decision makers and teams.

- Level 2: The career level is Similar to functional teams. Allows the formalization of interactions between Functional participants in the process, in which the business rules are adapted Of course.

- Level 3: The technical level links the activities / participants Similar to functional level, IT applications / services, as well as tasks Users (workflow). This level is implemented by architects and technical teams to the company.

2.3- Business Process Characteristics

From different definitions and multiplicity, the business process is dynamic in meeting the needs and expectations of customers, while changes in the market or contextual changes in the organization and from all these definitions we can now list the characteristics of a business process [4] [5]:

- For business operations internal services of the company and other external. Each service has different internal operations that remain within the company's boundaries. Internal processes are only used internal services. Some processes require interaction with others. These external processes are a problem of integration. From business to business. External operations use a range of services internal and external commercial process and these internal and external processes are the point of a series of steps planned to be implemented,

- One of its properties is also used in several functions.

-Specific actions by team members. The specified product, service or result that comes from the execution of the operation.

- The following process, student, or end-user of the result.

2.4-Benefits of Business Process

Integrating the techniques of business process into the information system of the companies has various benefits such as [9] [58]:

- Remove the grill at work.

-Increased profits.

-Improving the management of the company and regulating transactions between them.

-Reducing costs and increasing efficiency.

-Improving the quality of service.

-Increase adaptability, flexibility and agility.

-Reducing development and support costs.

-Reducing the risks associated with the application of new systems.

-The government's ability to support the government's activities in the abolish mediation by eliminating unnecessary intermediaries.

- Ease of distribution of knowledge and tasks and unifying the goal among the employees.

- Analysis of measurements and decision-making support.

- Automation of manual tasks.

- Increased customer focus.

2.5-Business Process types

The range of processes is what makes the business form an effective business model. To thrive business and improve processes and functions. The basic function of operations is to add value to the business. There are three different categories of processes that are used to enhance the business [36].

2.2.1. Processes Primary: These are the basic business processes through which the company delivers the final product to the customer.

2.2.2. Process support: Support processes do not add value directly to the end product but make an environment for initial operations to operate efficiently and effectively.

2.2.3-Management processes: management processes that govern operations, corporate governance and strategic management. These processes set targets and criteria that lead to the efficiency and effectiveness of the operation of primary and support operations.

2.6- Synthesis: Business Process types Comparing

The table below (Table 2.1) presents comparing types of business process (primary , support and management).

Types Different	Primary	Support	Management
Formelle	Not established.	established.	established.
Base	Considered to be the most important processes.	Supports the primary processes	Coordinates the activities of the primary and support processes.
Provide value to customers	Providing value.	Doesn't provide.	Doesn't provide.
direct contact with customers.	yes	No	yes

Table (1.1): *Business Process types Different.*

2.7-Analysis

Despite their common paradoxes, the third of them (Management processes) have another difference: it is Measures, monitors, and controls, Improve business processes efficacy and efficiency.

3- BUSINESS PROCESS MANAGEMENT (BPM):

3.1-Business process management (BPM) Definition [27]: is an organizational discipline where a company takes a step back and looks at all of these processes in total and individually. It analyzes the current state and identifies areas of improvement to create a more efficient and effective organization.

Business Process Management (BPM) is how a company creates edifies and analyzes the predictable processes that form the core of its business.

Every department in the company is responsible for taking data and converting it into something else.

The figure below (Figure 1.1) presents works Business Process Management [27].

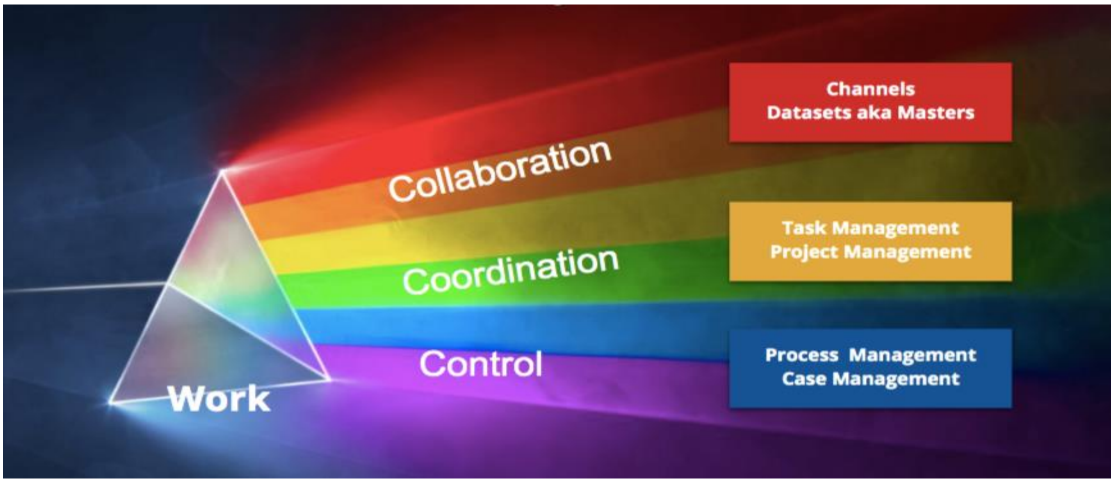


Figure 1.1: *Works Business Process Management.*

3.2-Business Process Management Features:

From definitions of business process and business process management definition, so we can say that a business process is a chain of activities that takes an input (of any form) adds value to it using resources, and provides output (product/service) that meets the company's objectives. It is triggered by internal or external events of the company. It can be devised into a sub-process and communicate with other processes [35] [34].

So we can conclusion same business process features:

-Represents a dynamic view of the organization (Has a goal).

-has an input and output.

- Adds value to goods or services.

- Roles that express the organization in the process.

- A transition function that controls the process.

- Resources, which can be means, information, or tools used by an activity.

- It has several functional modules.

- A chart of activities, representing the sequence of activities needed to implement the goal.

3.3-Business Process Management Life Cycle

The business process management lifecycle features five connected stages. Keep in mind you should follow them religiously, no matter what company or where in the world you are. The five stages are design, modeling, execution, monitoring, and optimization [25] [27] [26].

The figure below (Figure 1.2) presents a Business Process Management Life Cycle [25].

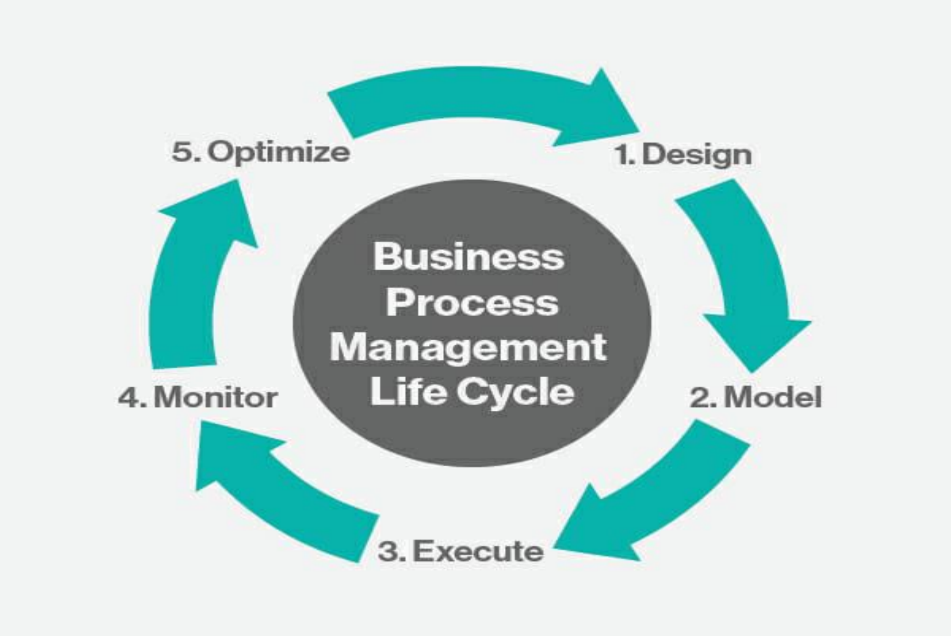


Figure 1.2: *Business Process Management Life Cycle.*

1- Design [27] [25]: we need observations from process analysis to be put to use in the design stage. At this point, you should have awareness of bottlenecks, lags, and delays in detail. The important question to answer is whether the process should be retained as is or redesigned to fix the issues identified. Based on the response, you may approach it in one of two ways:

- Continuous process improvement wherein the process is accepted in its current structure and issues are corrected one after another, or
- Redesign, wherein the entire process is remodeled in its entirety.

Once process modeling is complete, new procedures need to be approved. A deployment plan is created to ensure that relevant process performers are trained for the changes and transitions are smooth.

2- Model [27] [25]: Business process modeling is the process of taking the conceptual design of a business process and analyzing or improving it. You can see how processes might work under different conditions, and you will likely find that they don't remain the same under different circumstances. The goal here is to find out what will happen to each business process as conditions change. Implementing this process helps you plan a more efficient business process design so that you can understand and make positive changes.

3- Execute [27] [25]: The execution of the business process is about the age of discovered and extended business processes. The work process is performed manually or automatically. Manual implementation involves passing new operational guidelines to different departments and staff, what the process affects and what its effects are internally and externally.

However, BPM software groups such as BPMS are placed in the spirit of the business process. The program has been developed that allows the entire process to be specific in the computer language that can be performed directly by the computer. Process models can be operated through execution engines that automate processes directly from the model or when the step is too complex to be automated, the Process Design System (BPMN) provides the front already capable of human input.

4- Monitor [26] [27]: At this stage, whether it's performing BPM manually or automatically, you'll have to monitor activities and progress as the workflow progresses. This makes implementation and control go hand in hand. Once the process model is implemented, you should monitor the metrics (cycle time, defect rate, and productivity) to identify critical problems through and from the data to ensure the effectiveness of the exercise and to improve their connected processes, speed, quality and efficiency.

The principle on which this is based is simple: "Nothing is given, nothing is given; nothing is given. Everything is again." Nothing gives everything earned. And just so that you are aware, everyone who implements BPM must know the metrics to monitor, and this depends largely on your business objectives and reasons for implementing BPM.

5- Optimize [25] [27]: At this point, we know what works and what doesn't work. So to effectively improve your company's operations, we need to collect all performance information across all stages of the BPM lifecycle. This involves collecting data about what has worked and what has not worked, from design to model to monitoring. The big picture here is to ensure that mistakes are avoided and successes are repeated in order to reduce cost and maximize efficiency. No matter what the numbers look like, they don't tend to affect the design of the entire process. Instead, of more efficient workflow, focus on those key areas that need to be adjusted while leaving everything else to run as it is. The result of examples is usually modifying the current standard operating procedures.

In short, the BPM business process lifecycle is a continuous loop of design, model, implementation, control, and optimization. It is to ensure a smooth workflow in your company so that mistakes and successes are avoided repeated. And without a good knowledge of the workflow, practice our BPM.

To apply it correctly, make sure to look for a reliable or professional program.

3.4-Bisness Process Management Types

In this part, we will know types of business process management (orchestration, Choreography, collaborative).

3.4.1- Processes Orchestration

This type of business process management devises into 2 types other (private and public).And we see what difference between them.

A-Private Business processes

Private Business Processes are those internal to a specific organization. These Processes have been generally called BPM Processes. Another synonym typically used in the Web services area is the Orchestration of services.

There are two types of private Processes: executable and non-executable.

- **An executable Process** is a Process that has been modeled for being executed according to the defined BPMN execution semantics (during the development). Cycle of the Process, there will be stages in which the Process does not have enough detail to be “executable.”

The figure below (Figure 1.3) presents an example of a private business process [59].

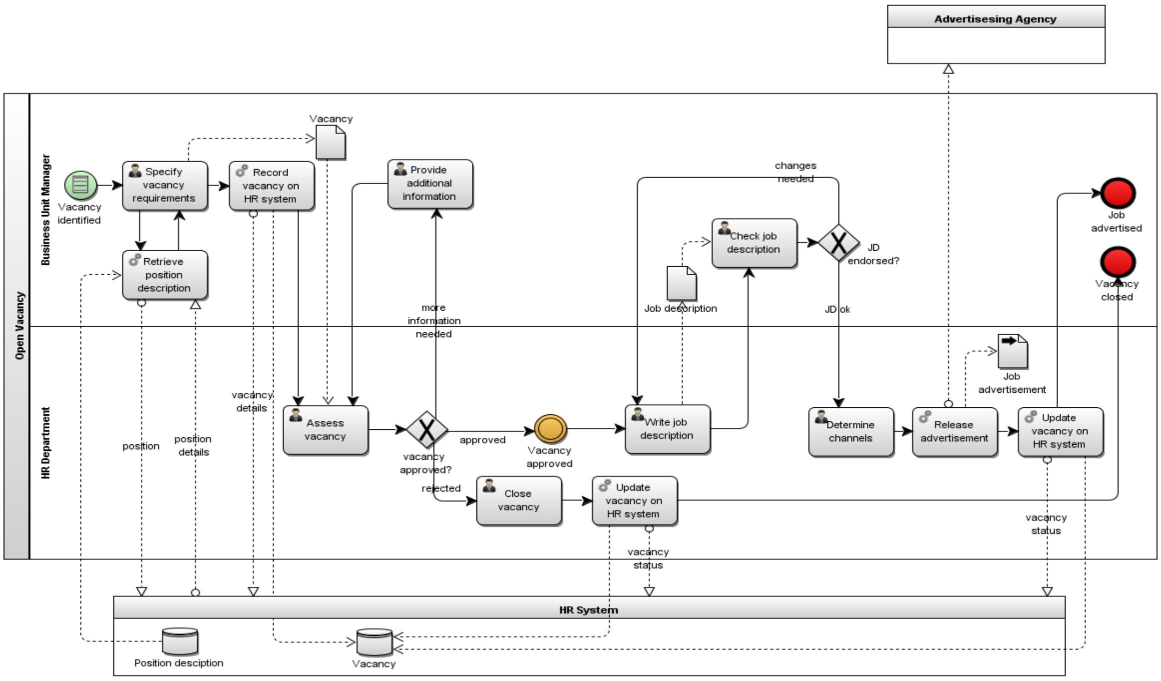


Figure1.3: *Example of Business Process executable.*

- **A non-executable Process** is a private Process that has been modeled for documenting Process behavior at a modeler-defined level of detail. Thus, information needed for execution, such as formal condition expressions are typically not included in a non-executable Process, defining the behavior of activities within a business process at a level of detail determined by the process modeler.

The figure below (Figure 1.4) presents an example of a public business process non-exactable [31].

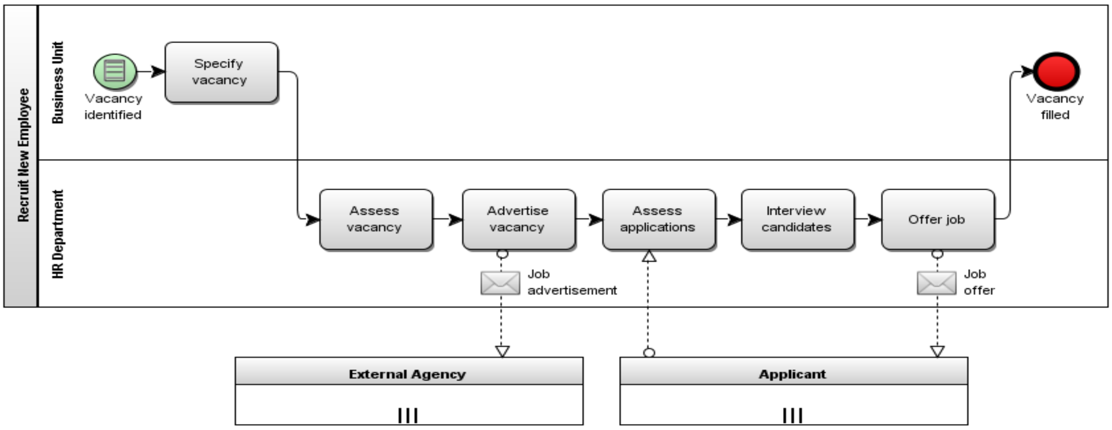


Figure1.4: *Example of Business Process non-exactable.*

B-Public Business Process:

A public business process [31] [32]: is a representation of the interaction between the internal business process and the participants outside the process. The internal activities of a private business process without any interaction with external participants will not be shown in this type of model, while a public process represents an interaction with another process or participant.

-B2B (Business-to-Business): processes between companies.

-B2C (Business-to-Consumer): processes between companies and customers.

-B2G (Business-to-Government): processes between companies and government agencies.

-C2C (Consumer-to-Consumer): processes between customers (online auction).

The figure below (Figure 1.5) presents an example of a public business process [31].

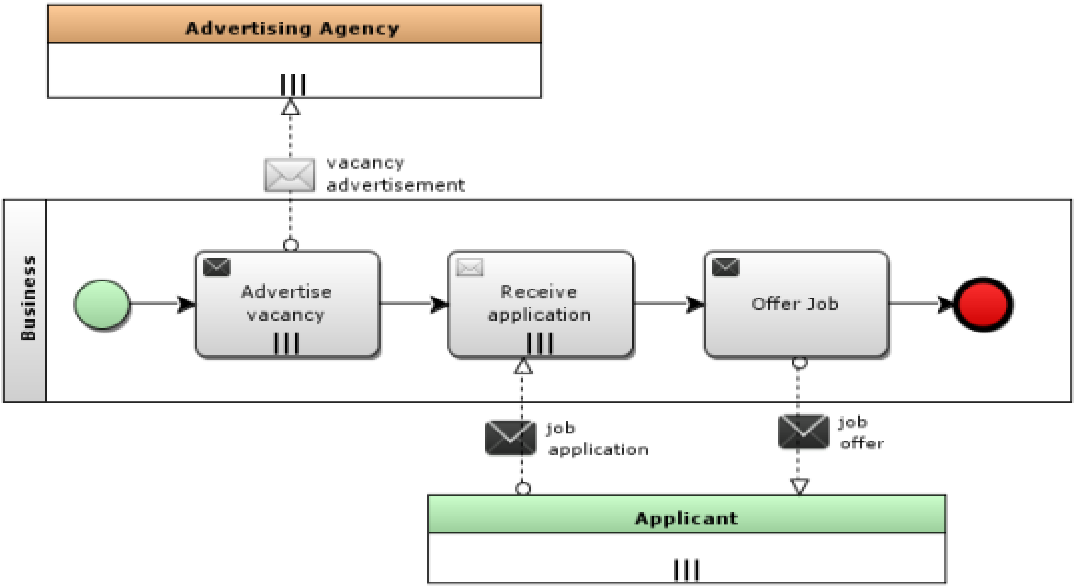


Figure1.5: *Example of public business process.*

3.4.2 - Choreographies Business process:

Choreography [6]: looks similar to a special business process because it consists of the network of activities, events and portals, however, the choreography. Varies in that activities are interactions that represent a group (one or more) of exchange of messases, which includes two or more participants. In addition, unlike a normal process, no central console, a responsible entity, or an observer of the process.

The figure below (Figure 1.6) presents an example of Choreographies business process [31].

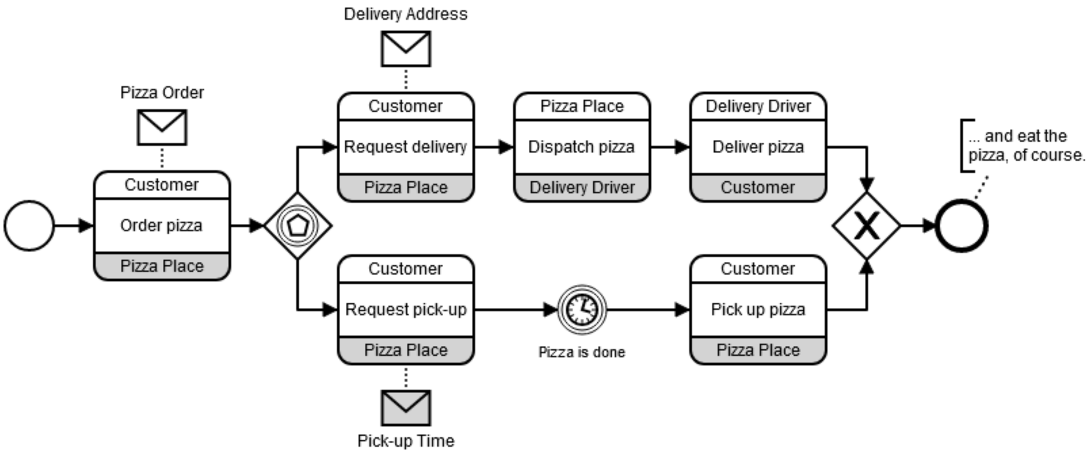


Figure1.6: Example of Choreographies business process.

3.4.3- Collaborative Business process.

Collaboration [6]: depicts the interactions between two or more business entities. Collaboration usually contains two or more Pools, representing the Participants in the Collaboration. The Message exchange between the Participants is shown by a Message Flow that connects two Pools (or the objects within the Pools). The Messages associated with the Message Flows can also be shown graphically.

The figure below (Figure 1.7) presents an example of Collaboration Business process [31].

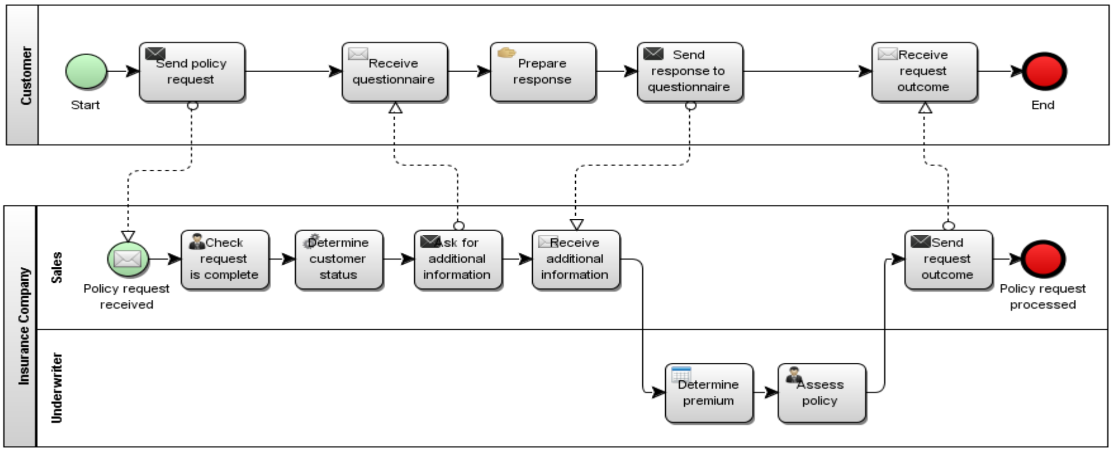


Figure 1.7: Example of Collaboration Business process.

3.4.5- Synthesis: Examples of public/private business processes

The table below (Table 1.2) presents our own study of some well-known enterprises and their public and private business processes.

Enterprise	Private Business Processes	Public Business Processes
Peugeot [11]	- Auto industry -Searching for new technologies and shapes -Increase new employees	-Types of cars and their features - Registration (B2C). -Request a trial. -order online -dialogue with your express (C2C).
Facebook [7]	- Add/delete/modify goods. - Add/delete/modify goods. - Hiring new employees -Updating	- Registration (B2C). -create post, stories. -send messages to friends (C2C).
Alibaba [8]	- Add/delete/modify goods. - Hiring new employees	- Deliver goods to customers - Customers can search for goods/ buy goods / review goods (B2C)
I phone **[33]**	-Smartphone industry -Hiring employees. -Paying employees.	- Deliver phone or other types to customers. - Registration (B2C).

***Table (1.2):** Examples of private and public business process .*

3.4.6-Analysis

Peugeot [11]: is a French car mark, this company was founded in 1810 and manufactured coffee mills and bicycles and his collaborate with car mark Citroën.

Facebook [7] : is defined as an online social networking website, we can create profiles and shared information with our friends like photos, quotes, videos, and all this we can send message to a theme.

Alibaba [8]: a listed Chinese company derives its income mainly from its online activities, including a public market aimed at facilitating exchanges between companies (whether international or Chinese), payment and retail platforms, a shopping search engine and cloud computing services. Its business is similar to Amazon, but it is more profitable.

I Phone [33]: a range of smart phones marketed by the American multinational company apple.

Thus, companies contain private business processes that are kept protected and safety.

3.4-Business process technologies

BPM technology helps enterprises achieve business agility, visibility, and efficiency via process automation, management, and optimization. There are four core functionalities that define BPM technology [16]:

3.4.1- workflow [37]: is a sequence of tasks that processes a set of data. Workflows occur across every kind of business and industry. Anytime data is passed between humans and/or systems, a workflow is created. Workflows are the paths that describe how something goes from being undone to done or raw to process.

There are three types of workflow [14]:

1-The workflow of the operation: The process workflow occurs when the task set is predictable and replicated. That means exactly what the course of the operation is, and if he doesn't know the course.

Process workflows are set up to handle an unlimited number of items that pass through them. An example is the processing of a purchase order approval.

2-The workflow of the case: In the status workflow, you don't know the path required from the beginning. You should collect more data to reveal the path itself, such as support tickets.

Similar to the workflow process, workflow status can handle any number of items, although it relies on a human or intelligent robot to distinguish the right way.

3-Project workflow: is only good for one item. Another version of websites may not be released for a long time and will probably not follow the exact same path.

Most of the resources you will find online only refer to the workflow in the sense of practical workflow, but the other two are just as important to consider a lot of work around the office falls into these two categories.

3.4.2-Services web [38]: A web service is a standalone application or component, example: a semantically well-defined feature, identified by a uniform resource identifier Uri. It represents a standardized way of integrating applications based on Web using XML standards SOAP, WSDL, UDDI and Internet transport protocols. XML is used to represent the data, SOAP to transport data, WSDL to describe available services, and UDDI to list different service providers and available services.

The figure below (Figure 1.8) presents a Pile of web service protocols [39].

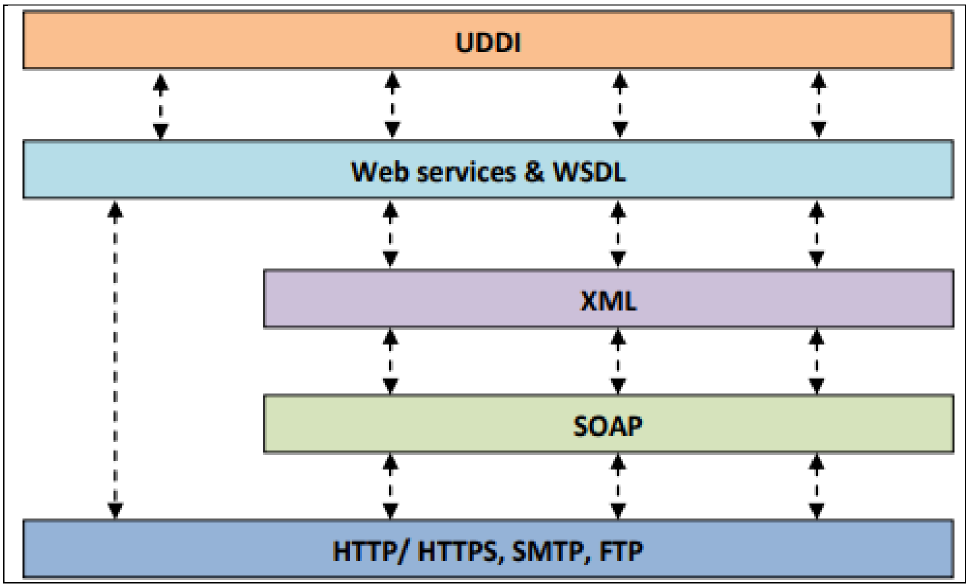


Figure 1.8: *Pile of web service protocols.*

3.4.3-The BPM engine:

The BPM engine replaces workflow engines. He is in charge of choreographing IS services. Today, BPM delivery engines meet the standard BPEL "Business Process

Execution Language". The general use of this standard allows for interoperability between the various tools in the market.

3.4.4- Automation of business processes*:* Business process automation involves using technology to perform recurring tasks or processes in a company where manual tasks can be replaced. The goal is to minimize costs, increase efficiency and optimize processes.

4- BUSINESS PROCESS MODLING

4.1- Business Process Modeling [10][12]: this technique is used for analysis, modeling, implementation and excited or used in an organization to get a clear picture of the current procedure and find new ways to bring an improvement to the ongoing processes.

It is an organizational service that systematically performs these daily tasks, often unstructured or repulsive, to improve your business process and increase efficiency [29].

Chaotic processes damage businesses and lead to :(Lost time, more mistakes, increased blame, Lack of data, frustrated employees).

Applying business operations management, organizations can optimize their operations and maintain all aspects of operations.

4.2-Benefits of Modeling

Business process changes a lot and develops for the better through planning and modeling, so that makes it easier for users to do business, and on this has many benefits, we represent them in suit[13][17]:

-Detect errors and reduce system flaws explore alternatives earlier

-Simulating system solutions without creating code.

- The picture and objectives of the entire organizational processes of each individual of the company, so everyone will have the same idea and all of them on the same page.

- Eliminate repetition.

- Best compliance with industry standards.

-Get a clear picture of the entire organizational processes

-Identify processes using workflows and understand the tasks being carried out in a graphical representation.

-Make important business decisions after having a thorough idea of what is required and what is not.

-Understand the need for automation and stop redundant tasks from occupying your workforce’s precious time.

-Create a better workflow process for the business to perform better and pave way for future improvement or expansion.

4.3- Business Process Modeling Classification

Multilingual and multiple functions each one of them has a type to study, so we can classify languages modeling into several categories and one of them is (formal, informal and semi-formal).

a- Business process Modeling Formal:

1-Petri nets [15]: are specific types of modeling constructs useful in data analysis, simulations, business process modeling and other scenarios. This type of mathematical construct can help to plan workflows or present data on complicated systems.
Petri nets use elements like places, transitions and gates to describe complex procedures and model the workings of a system. Tokens and marking systems can show movement through these systems. Much of the model can be delivered in classic mathematical notation, or specific drawings conforming to the syntax and makeup of the Petri net model.

Petri nets have many practical applications in technology. One big example is where Petri nets are used in business process analysis. Another example is where tools like sequential function charts (SFCs) in programmatic structures are useful for the observation and implementation of systems.

2-Pi-Calculus [21]: In theoretical computer science, the *π*-calculus (or pi-calculus) is a process calculus. The *π*-calculus allows channel names to be communicated along the channels themselves, and in this way it is able to describe concurrent computations whose network configuration may change during the computation.

3-XML Process Definition Language (XPDL) [24] : is a format standardized by the Workflow Management Coalition (WfMC) to interchange business process definitions between different workflow products, i.e. between different modeling tools and management suites. XPDL defines an XML schema for specifying the declarative part of workflow / business process.

b- Business Process Modeling Informal:

1-Business Process Definition Met model (BPDM) [22]: is a standard definition of concepts used to express business process models (a met model), adopted by the OMG (Object Management Group). Met models define concepts, relationships, and semantics for exchange of user models between different modeling tools. The exchange format is defined by XSD (XML Schema) and XMI (XML for Metadata Interchange), a specification for transformation of OMG met models to XML.

2-Business Process Modeling Notation (BPMN) [23]: is a graphical representation for specifying business processes in a model. Originally developed by the Business Process Management Initiative (BPMI), at which point the name was amended to Business Process Model *and* Notation to reflect the introduction of execution semantics, which were introduced alongside the existing notational and diagramming elements. Though it is an OMG specification.

3-BPEL4WS [Business Process Execution Language for web service] [10] is a specification proposed jointly by IBM, Microsoft, and BEA, and represents a convergence of the ideas initially proposed by the languages XLANG and WSFL. Which describe business processes in the form of Web services, and how they are interconnected to perform particular tasks.BPEL4WS is an effort to standardize the composition of web services. It is also a language for defining and managing activities of a business process. This language makes it possible to describe protocols of interactions and collaborations between the web services on which the process is based.

c- Business process Modeling Semi-formal:

1-Integrated Definition Method 3 (IDEF3) [20]: is a business process modeling method complementary to IDEF0. The IDEF3 method is a scenario-driven process flow description capture method intended to capture the knowledge about how a particular system works.

The IDEF3 method provides modes to represent both:

- Process Flow Descriptions to capture the relationships between actions within the context of a specific scenario.
- Object State Transition to capture the description of the allowable states and conditions.

This method is part of the IDEF family of modeling languages in the field of systems and software engineering.

2-Unified Modeling Language (UML) [19]: is a modeling language mainly used for specification, visualization, development and documenting of software systems. But business professionals have adapted it as a powerful business process modeling technique. With 14 different UML diagram types, it offers a flexible and powerful way to visualize almost any business process. They are typically used for modeling the detailed logic of a business process. We present the important diagrams as:

Structure Diagrams

-**Class diagram:** Describe the structure of the system by showing the system's classes, their attributes, operations (or methods), and the relationships between objects.

-**Component diagram:** shows the structural relationship of software system components. These are mainly used to deal with complex systems with many components. Use interfaces to communicate between components. Structure chart

-**Timing diagram:** Describe different scenarios of using the system.

-**Activity diagram:** represents a step-by-step business workflow and operational components

-**Communication diagram**: This allows highlighting the message exchanges between objects.

-**State-transitions diagram**: Represents states and state transitions.

-**Global interaction diagram**: Provides an overview and nodes representing communication diagrams.

-**Time diagram**: This is to describe the interactions between objects with strong temporal constraints.

4.4-Synthesis: Comparing the different modeling tools

We can be classified langue Business Process Modeling in three paradigms (formal, semiformal, and informal), the following (Table 1.3) presents a simple comparison between the different modeling tools mentioned above.

	Language	Domain
Formal	**Petri Net**	-used analysis, simulations, business process modeling -Tokens and marking systems can show movement through these systems -delivered in classic mathematical notation, or in specific drawings
	Pi calculus	-Used to describe concurrent computations whose network configuration may change during the computation and calculus expiration formal.
	XPDL	-used to specify the declarative part of workflow / business process.
Informal	**BPMN**	-Used in the specification and publication of business processes. -Used in Systems that change dynamically.
	BPDM	-Used in process design (or redesign) during the business Process management stages when a company analyzes studies and remains certain

		business processes.
	BPEL4WS	-Used in the specification and publication of business processes. -Used in Systems that change dynamically.
Semi-formal	**UML**	- used for specification, visualization, development and documenting of software systems. - used for modeling the detailed logic of a business process
	IDEF3	- Flow Descriptions to capture the relationships between actions within the context of a specific scenario. - Transition to capture the description of the allowable states and conditions. -Modeling in the field of systems and software engineering.

Table (1.3): *Comparing the different modeling tools.*

4.5-Analysis:

All types are specifications for a commercial process but each type is studied in some way, the formal method depends on the purely mathematical method and has special methods for its electronic processing, but this is a bit difficult for the management workers.

Semi-formal is heavily dependent on graphic language and using formal models that combine graphic languages with official connotations, which makes it possible to better understand the performance of the system, and therefore it is a good way to control its complexity.

As for the informal is a good specification where it combines the models drawn to our electronic shoes in an easy way and this leads to the ease of dealing with companies and simplifies the way of work.

5. SPECIFICATION

5.1-Rdp color [46]: is a graphic structure with a set of places and transitions, connected by oriented arcs, possibly carrying weight. These arcs are links between place and transition or between transitions and place exclusively. In this structure move tokens (or marks) that appear in the squares and are likely to make transitions according to certain criteria franchisability and crossing.

The figure below (Figure 1.9) presents a graphic structure Rdp color [18].

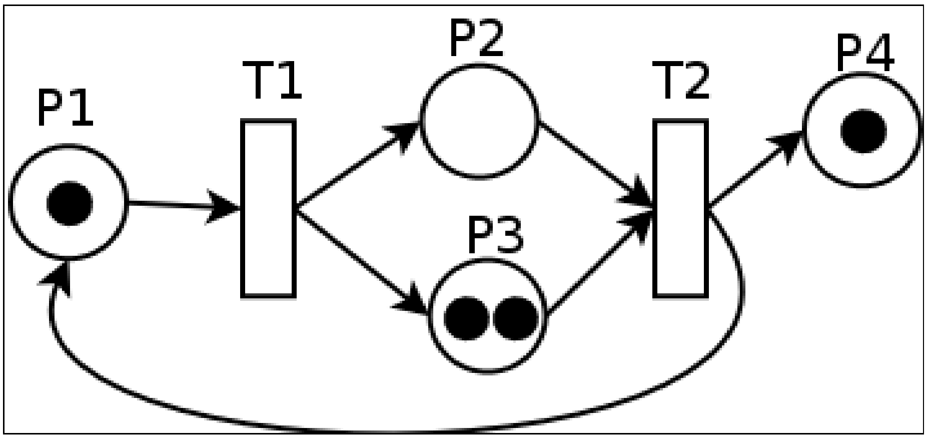


Figure 1.9: *Graphic structure Rdp color.*

5.2-AEF [45]: A finished-state automaton is the simplest abstract machine (program) that recognizes regular languages; it does not use any memory.

An AEF is formally defined by a quintuple (X, Q, I, F, δ) where:

-X is a finite set of symbols (the letters of the alphabet);

-Q is a finite set of states;

-I is the initial state, I∈Q;

-F is all final states, F⊂Q;

-δ is the transition function, defined from Q×X in Q.

The figure below (Figure 1.10) presents Automaton (AEF) [30].

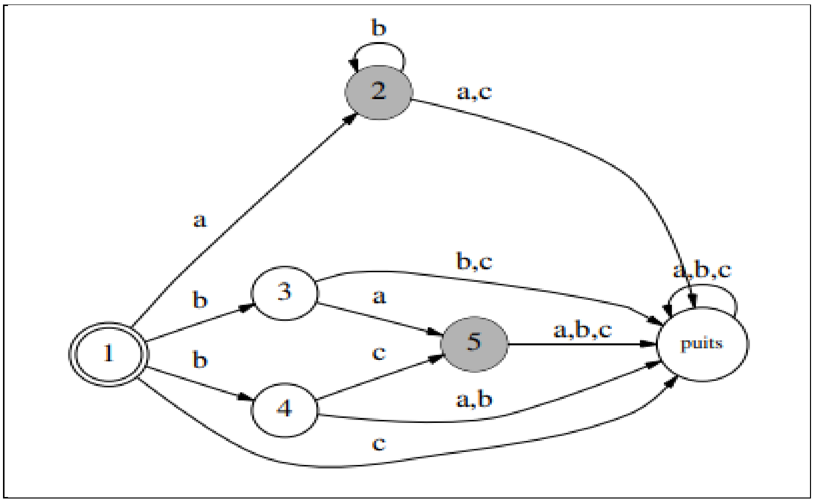


Figure 1.11: Automaton (AEF).

5.3-Timed Automaton (TA) [43]: is a finite directed graph annotated with conditions over and resets of non–negative real-valued clocks and a system is modeled as a collection of finite state machines and a finite set of clocks. In the standard scheme, the clocks are synchronized and can be reset by the transition from one state to another. Clocks are also used to guard transitions. Within TA, time is continuous but the clock's statements are usually restricted to using integer values. Time is never negative as the clocks can only be resets to 0.

The figure below (Figure 1.11) presents a Timed Automaton [28].

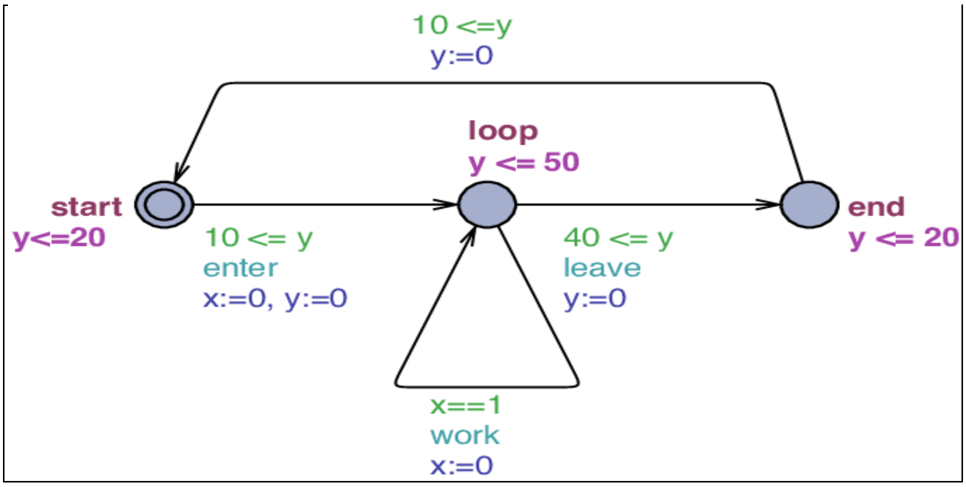


Figure 1.11: Timed Automaton.

5.4-Logic Temporal [44]: is any system of rules and symbolism for representing, and reasoning about, propositions qualified in terms of time. It is sometimes also used to refer to tense logic, a modal logic-based system of temporal logic. and it has found an important application in formal verification, where it is used to state requirements of hardware or software systems. It has two types of specifications (LTL, CTL).

5.5-Model Checking (MC) [42]: is a formal verification technique that enables exhaustive and automatic checking of whether or not a model meets a given specification. Thus, applying MC to Business Process (BP) verification helps to solve problems as bottlenecks and deadlocks. To apply the MC technique, the BPs needs to be described in a formal language.

5.6-Synthesis

We can be comparing the specification in three formal paradigm (formalism, Verification/Validation, domain), the following (Table 1.4) presents a simple comparison between the different specification mentioned above.

models **criteria**	**Formalism**	**Verification/Validation**	**Domain**
Timed Automaton (TA) [43]	Formalism relatively simple to manipulate but possessing expressiveness necessary for the modeling competing systems Delay.	- - Check a specification CTL by model-checking. -Validating the proper functioning a simulation system.	Computer science, mechanics, The automotive industry, The Wood Industry, Machine tool, Product Control, Product shaping finished, Conditioning, pallet packaging after packaging.

Logic Temporal [44]	System deduction based on Classical logic to which he added operations.	To check to transform informal specifications axioms using logic CTL and TCTL.	Computer science, the field of formal verification.

Table (1.4): *Synthesis of formal modeling.*

5.7-Analyse

In the field of software engineering, models are established, including models used in automation, computer science, operational research, and economics it is also a good way to control its complexity. For our work, we have treated two forms of models that are formal models and semi-formal models. We are focused on formal models: AEFs, logic time.

The timed finite state automatons is a relatively simple formalism to handle but having the expressiveness needed to model competing systems and for which there are powerful model-checking and simulation tools. As long as checking the time specifications of a system requires a formalism manipulates time constraints, time automatons is a formalism adapted to such constraints, such as the temporal properties of a system can be modeled (all grades and invariants) and/or the test (checks properties that are presented by a specification formalism such as the CTL time logic using a verification tool).

. We conclude that formal models are needed to model a system where transforms informal specifications to formal specifications.

6-CONCLUSION

In this chapter, we learned the difference between business operations and managing business operations. We mentioned the comparison between different type of business processes (primary, support, and management), their advantage, the life cycle of our technologies, and in the end, we mentioned different special languages of modeling such as UML, BPMN, BPMN4WS, in the end, give us an example with different types diagrams of BPMN.

The next chapter provides analysis and the concept of our system (Assembly line system car), it offers the use of our case studies and their public and private business processes, then represent them using UML diagrams and temporal SMA (stat Machine Automat).

CHAPITRE 02 : ANALYSIS AND CONCEPTION

CHAPITER 02: ANALYSIS AND CONCEPTION

1-INTRODUCTION

An assembly line car is a manufacturing process (often called a progressive assembly) in which parts (usually interchangeable parts) are added as the semi-finished assembly moves from workstation to workstation where the parts are added in sequence until the final assembly is produced [52].

However, these systems are considerate as very critic in time (synchronization), in risk (robots, automatic arms) and in cost (expensive maintenance). So, main factors need to simulate all system's comportment before implant the real and the hard system.

For this, this chapter defines first the different external and internal business processes of an assembly line of car, using model UML [unified modeling language] by making the necessary diagrams that help us to understand and to analyze this system and how it works in a detailed manner. Second, a timed finished state automaton model is elaborated by UPPAL and verifies them by CLT.

2. ANALYSIS

In the first, we chose a business process assembly line for study case:

2.1-STUDY CASE: CAR ASSEMBLY LINE

In this section, we present our study case as a cars assembly line for several motivations.

2.1.1 Cars assembly line Definition

An assembly line cars continent two principle business Processes (BP): robotic arms and supply chain:

Robotic arms [50]: machines that are programmed to execute a specific task or job quickly, efficiently, and extremely accurately. Generally, Motor-driven, they're most often used for the rapid, consistent performance of heavy and/or highly repetitive procedures over extended periods, and are especially valued in the industrial production, manufacturing, machining and assembly sectors.

The figure below (Figure 2.1) represents the robot arm [52].

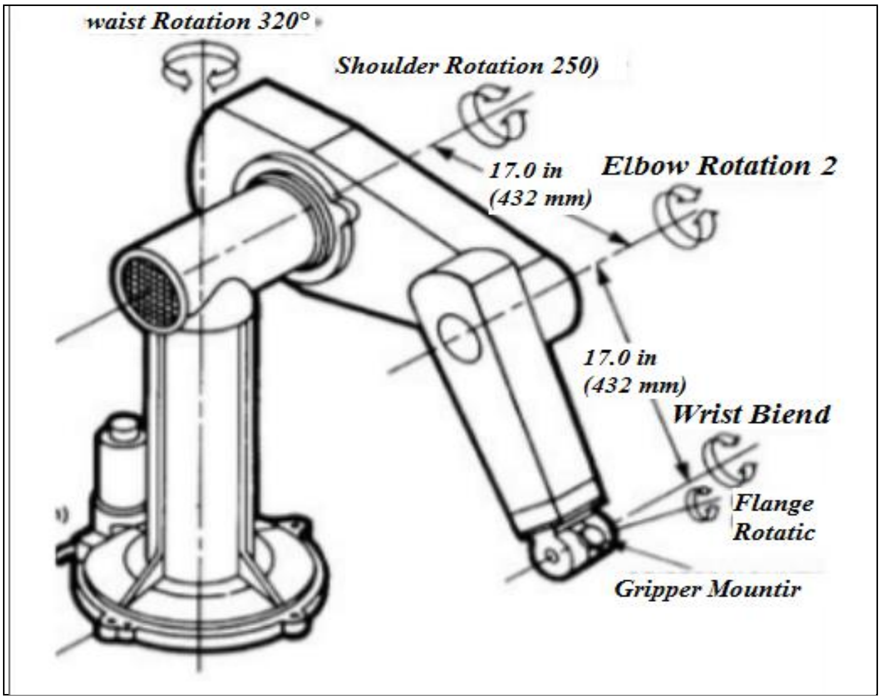


Figure 2.1: *The robot arm.*

Supply chain [53]: operates on three levels, strategic, tactical and operational. While the strategic approach is generally about improving network resources such as network design, location, facility count determination, etc., tactical decisions deal with mid-term, including production levels in all factories, assembly policy, inventory levels and lot sizes.

The figure below (Figure 2.2) represents a supply chain [56].

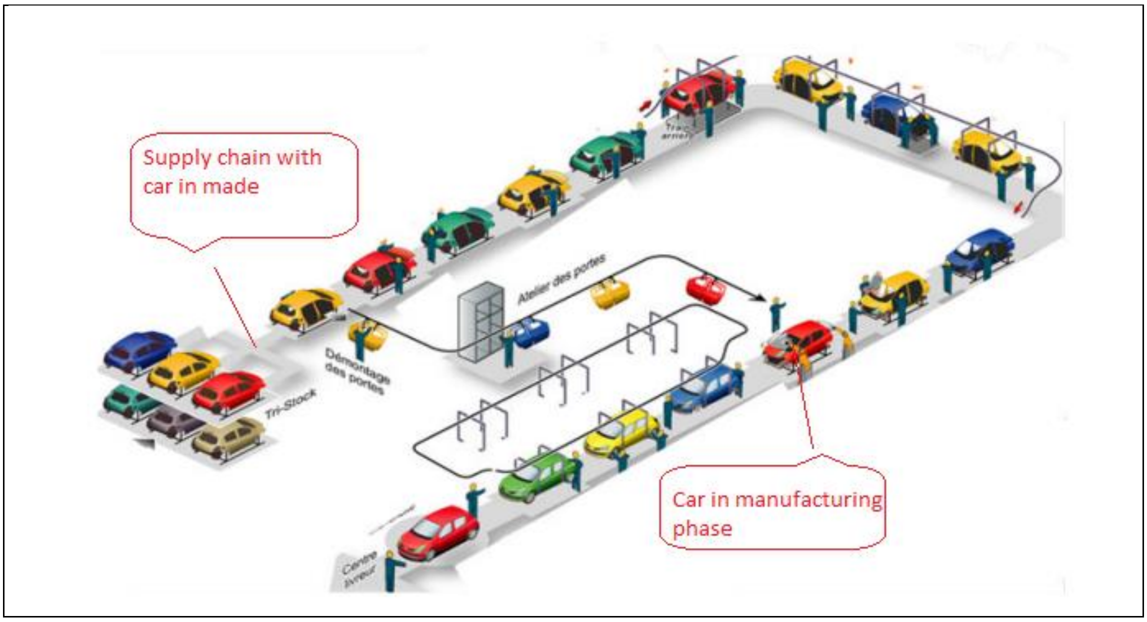


Figure 2.2: *The supply chain.*

2.1.2- Motivations

An assembly line car has been chosen as a study case of a modernized information system, and this choice is backed by the following motivations:

- For the first time, an auto-makeup application has been made with a view to increasing production.

-It's easy for us to find the malaise quickly without wasting it.

- Easy to distinguish between its businesses processes which leads to a better understanding of how our modeling will be.

- Work is clear and this easy makes it for us to business process synchronization components according to time.

- Its characteristics are the most suitable for the aspect of our modeling.

2.1.3-The different stages of building a car by robotic bras:

For increase the number of car production and speed of work, we must allocate every robotic arm in the field of build. The different stages of assembly cars:

A-Stamping: To produce the bodies, the steel sheets used are sent by the steel company in the form of large rollers. The steel sheet is cut according to the size of the piece and is folded and cut into a stamping machine designed to make important parts such as car doors, roofs and hoods [54]**.**

B-Assembly Stamped (Welding): After the stamped parts are made, they are assembled to form the main body. The sheet metal department is responsible for this task. Steel plate parts are assembled by spot welding and laser. For the assembly of aluminum parts (such as hoods and doors), the technician used is riveting or riveting. It should be noted that the assembly is usually performed by a robot [55].

C-Painting: The paint follows the assembly. Since a single layer is not enough to prevent corrosion, in the paint department, the shell has to go through several layers. In order to ensure sound insulation and at the same time ensures the waterproofs of the vehicle, putty will be applied first. The base layer of baking varnish and varnish is applied only after the primer is applied. These give the cash register its final appearance.

To accomplish this, wax is injected into the hollow body of the housing. Again, the purpose is to ensure corrosion protection [54] [55].

D-The editing: After coating and processing all the necessary equipment: engine,

cockpit, dashboard, headlights, seats, flash, interior trim, wheels, etc. Assembly is still the most difficult step and requires concentration. Of course, before assembly, the management must make sure that all the components of the car are ready and compatible, to avoid any mistakes and to build the car to the highest-quality requirements and above all to comply with safety standards.
The car is installed by workers in collaboration with advanced machines as well as robotic arms [54] [55].

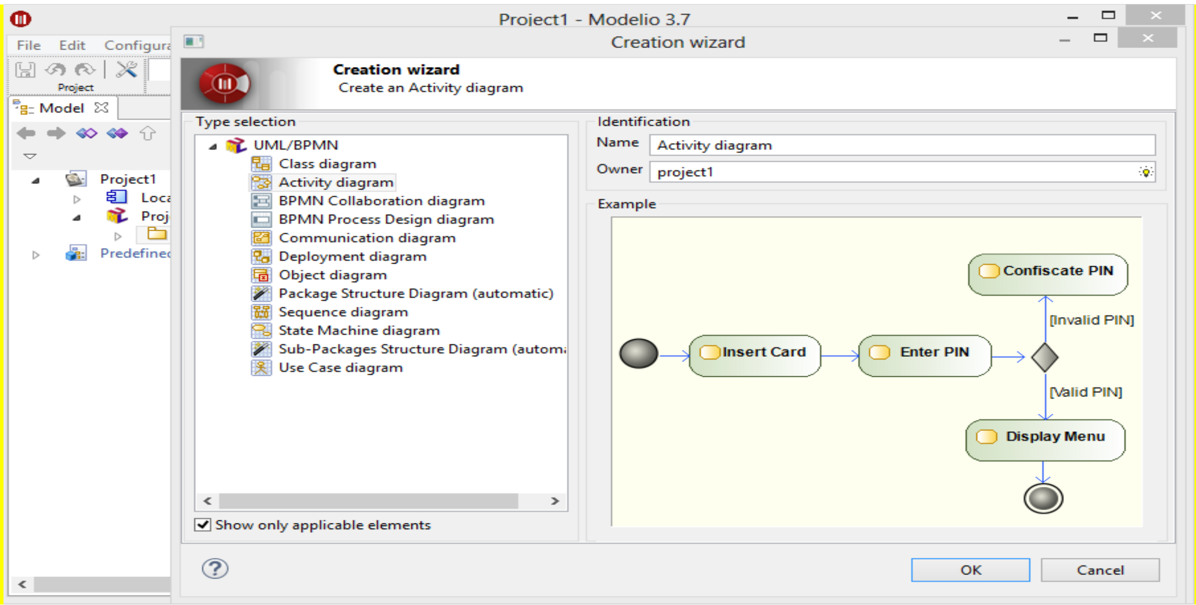
3. SYSTEM CONCEPTION

This section provides to modeling the system using UML models, to elaborate the use case diagram, class diagram, sequence diagrams and activity diagrams.

3.1-Modeling Software Tool

Modelio: is an open source modeling environment which can be extended through modules to add functionalities and services, analysts, designers, business and system architects such as UML and BPMN [47][48].

The figure below (Figure 2.3) represents the graphical user interface of Modelio.



Creation wizard
Create an Activity diagram

Name: Activity diagram
Owner: project1

☑ Show only applicable elements

Figure 2.3: *Modelio GUI.*

3.2 Use Case Diagram:

In this part we present the use case diagram of an Assembly line car with Chain which represents the actions that the different actors can do: technician and his relation seep between robotic arms and supply chain, see (figure 2.4):

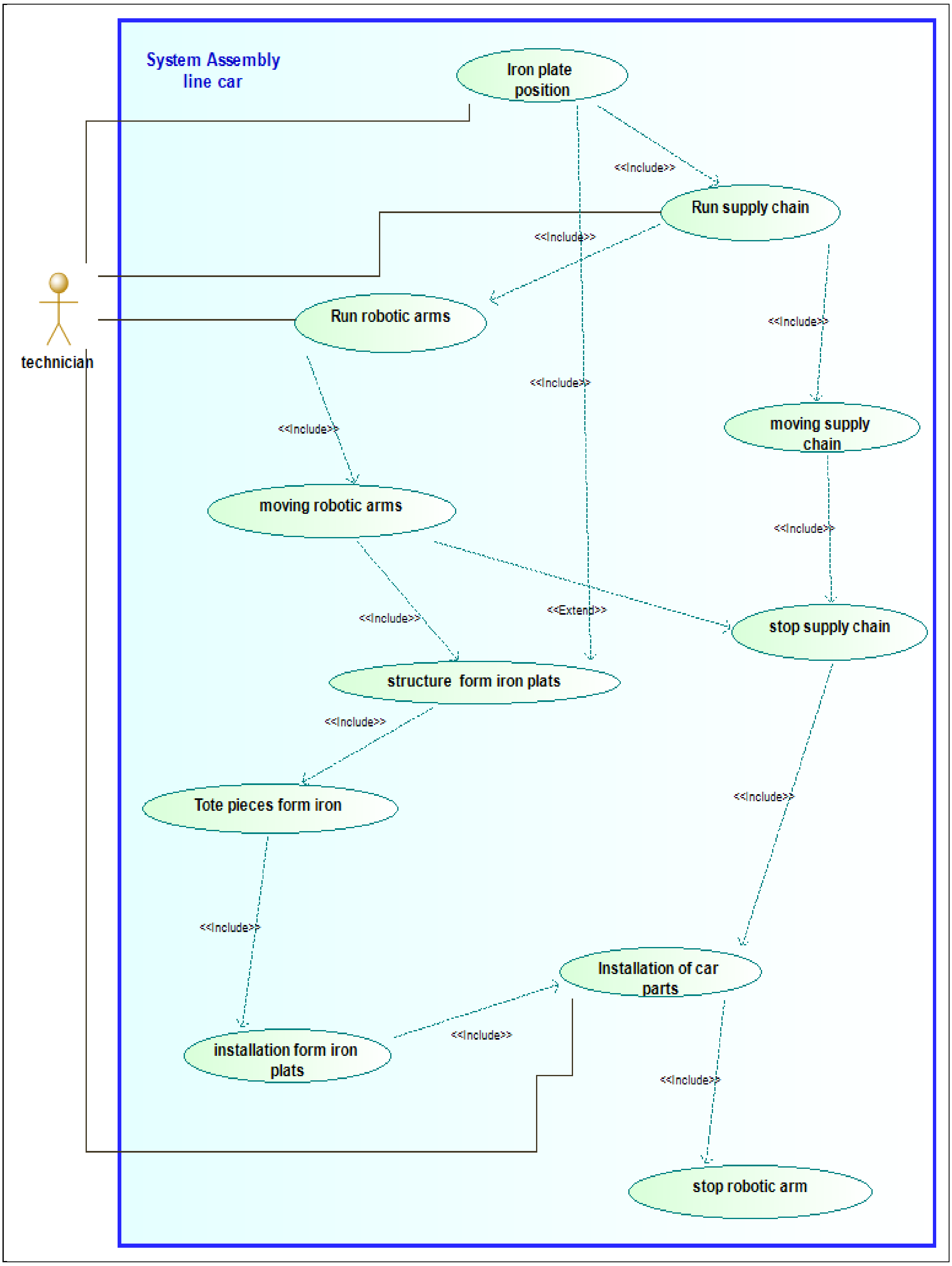


Figure 2.4: Use case diagram.

The table below (Table 2.1) presents a description of and its use cases in our system (An Assembly line car).

Actor	**Use case**	**Description**
Technician	- Iron plate position - Run supply chain - Run robotic arms - Installation of car parts	**-** Technician is a primary actor in our assembly line car and finally because he starting move of robotic arms, supply chain and in the fin of operation, he completes to the installation of another party that car need.
	- Moving robotic arms - Structure form iron plats - Tote pieces form iron - Installation form iron plats - Stop robotic arm	- Robotic arm must be made from irons parts.
	- Moving supply chain - Stop supply chain	-He change spats between robotic arms.

Table 2.1: *Actor and use cases description.*

3.3 Class Diagram:

In this part, we present the class diagram of an Assembly line car which is considered the most important concept in our design as set of tables (attributes, methods) because they will be used after in building our data base (Figure 2.5).

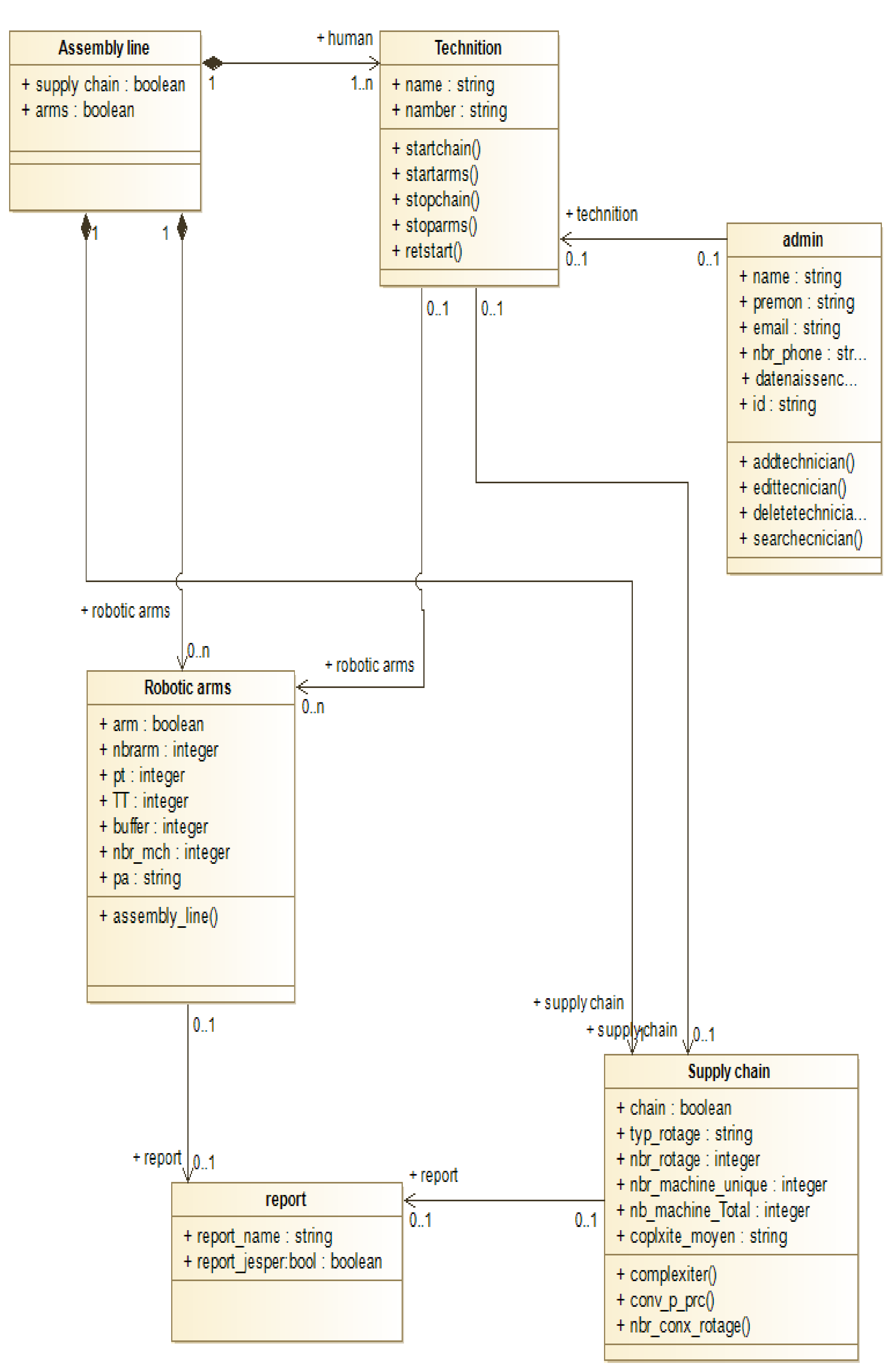


Figure 2.5: *Class diagram.*

The table below (Table 2.2) explains each class in the class diagram shown above (Figure 2.4).

Class	**Analysis**
Technician	Contains 2 attributes: email and password, and methods: start_chain(), start_arms(), stop_chain(), stop_arms(), restart().
Assembly line	It contains 2 attributes: supply_chain, arms, It is related by composition with the class Human, Robotic arms and Supply chain (contains from 1 to n number of human, and from 0 to n Robotic arms, and 0 to 1 of Supply chain) and also have from 1 to n number of admin.
Admin	It contains attributes: id ,nom , prenom ,nbrphone ,email ,datenaissene, It is related with class technician, (contains from 1 to n number of technicien) and also have from 1 to n number of admin. methodes: add_technician(), edit_technician(), delete_technician(), seartch_technician().
Robotic arms	Contains attributes: arm , nbrarm . Methods: startarm(), restartarm(),stoparm();
Supply chain	Contains attributes :chain ,type_rotage ,nbr_rotage, nbr_machine_unique, nbr_machine_totalncomplexite_moyen. methods: complexiter(), conv_produit_produit(), nbr_conx_rotage();
Report	Contains 2 attributes: report_name,report_jasper.

Table (2.2): *Class diagram analysis.*

3.4 Sequence diagrams:

Specifying the operation of each component of our system is made by the sequence diagram that shows its dynamic behavior.

3.4.1: Sequence diagrams of authentication: Figure 5 shows the how the authentication activity works for all actors: user, technician:

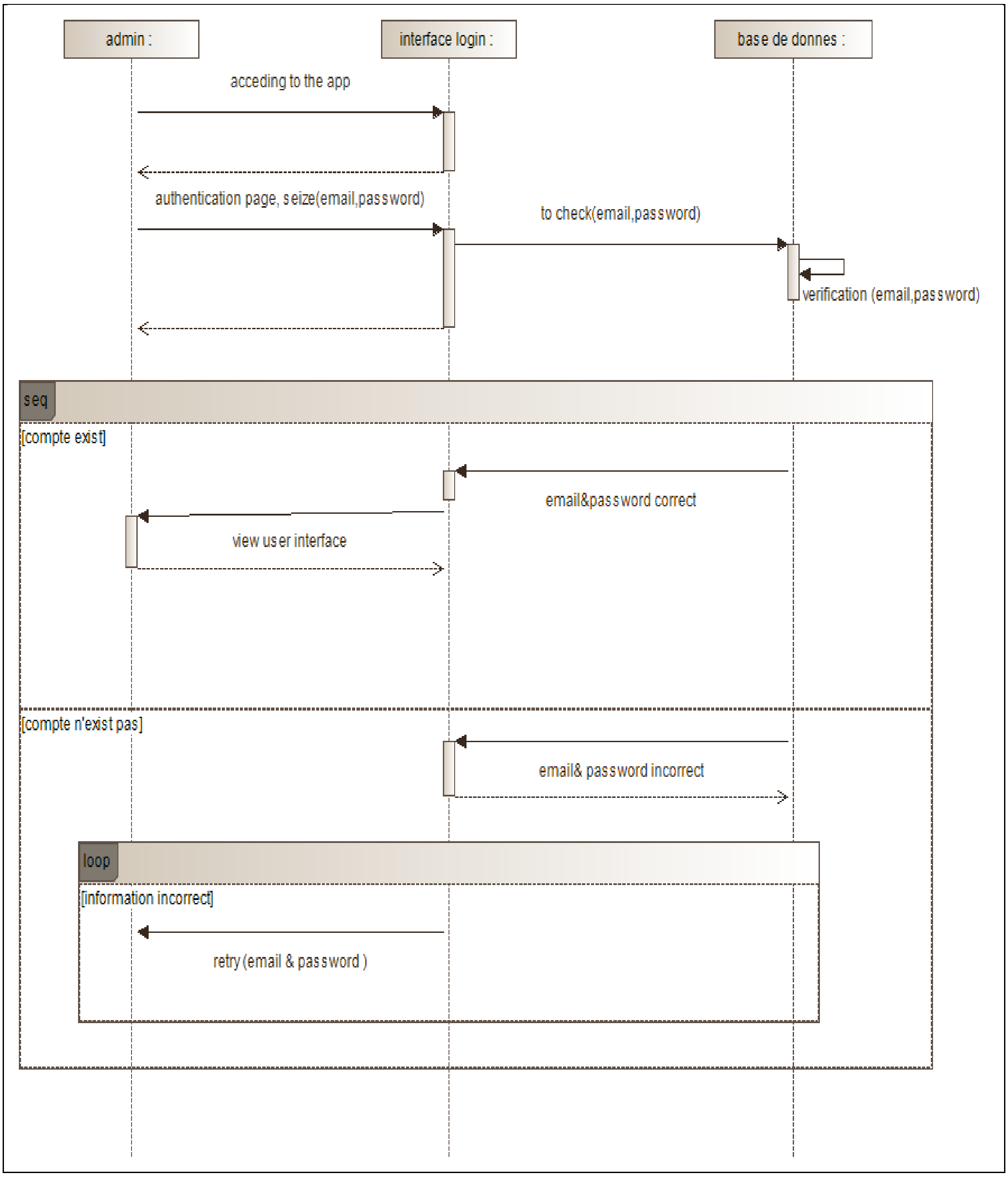


Figure 2.6: *Sequence diagrams of authentication.*

3.4.2: Sequence diagrams of addition: the figure 6 shows the diagram for adding the technician:

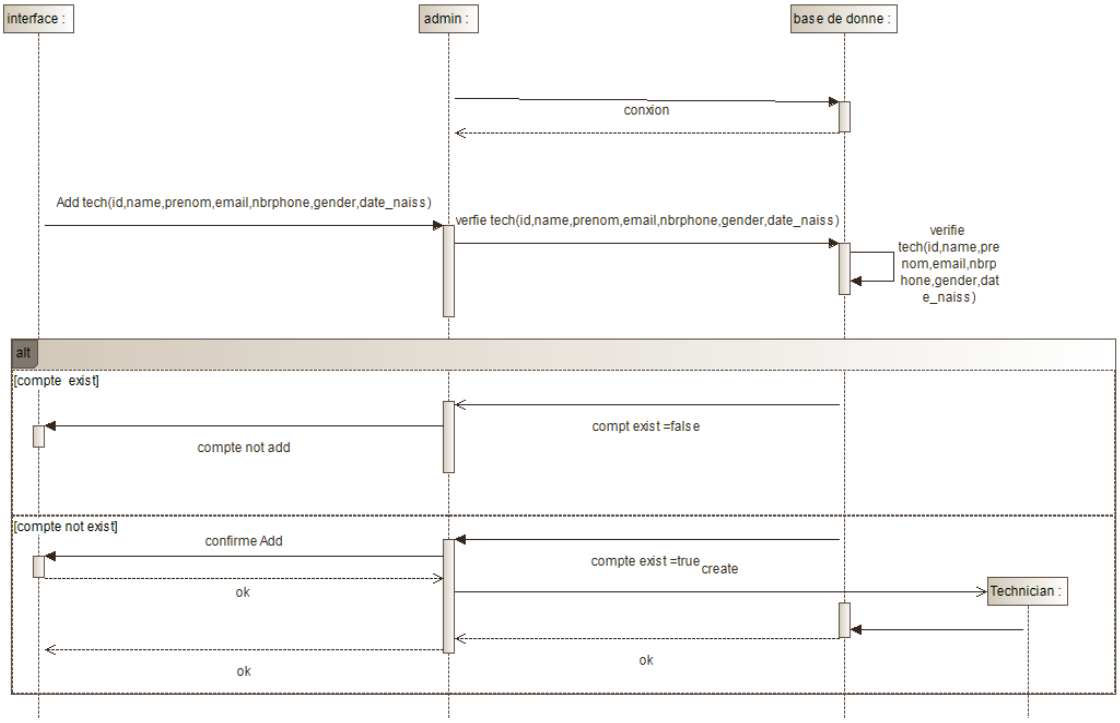

interface :
admin :
base de donne :
conxion
Add tech(id,name,prenom,email,nbrphone,gender,date_naiss)
verfie tech(id,name,prenom,email,nbrphone,gender,date_naiss)
verifie tech(id,name,prenom,email,nbrphone,gender,date_naiss)
alt
[compte exist]
compt exist =false
compte not add
[compte not exist]
confirme Add
compte exist =true
create
Technician :
ok
ok
ok

Figure 2.7: Sequence diagrams of add technician.

3.4.3: Sequence diagrams of edition (modify): the figure 7 shows the diagram for changing the technician account:

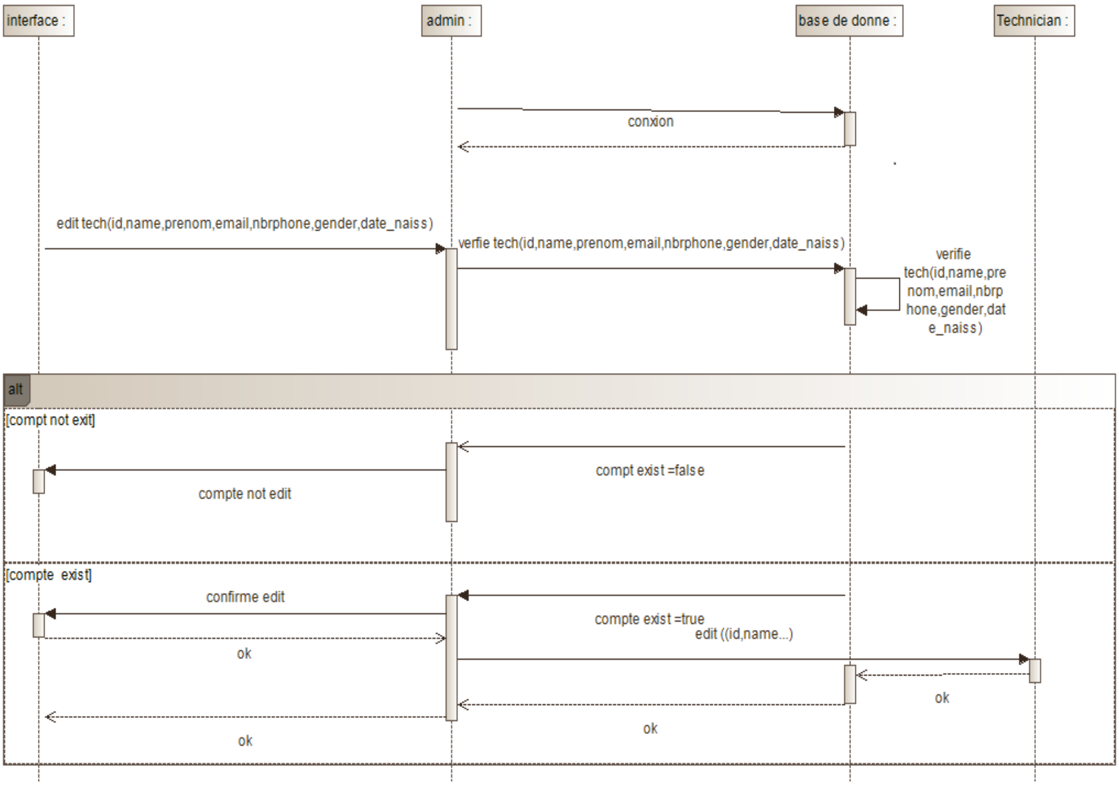

interface :
admin :
base de donne :
Technician :
conxion
edit tech(id,name,prenom,email,nbrphone,gender,date_naiss)
verfie tech(id,name,prenom,email,nbrphone,gender,date_naiss)
verifie tech(id,name,prenom,email,nbrphone,gender,date_naiss)
alt
[compt not exit]
compt exist =false
compte not edit
[compte exist]
confirme edit
compte exist =true
edit ((id,name...)
ok
ok
ok
ok

Figure 2.8: Sequence diagrams of modify in information of technician.

3.4.4: Sequence diagrams of delete: the figure shows the diagram for delete the technician account:

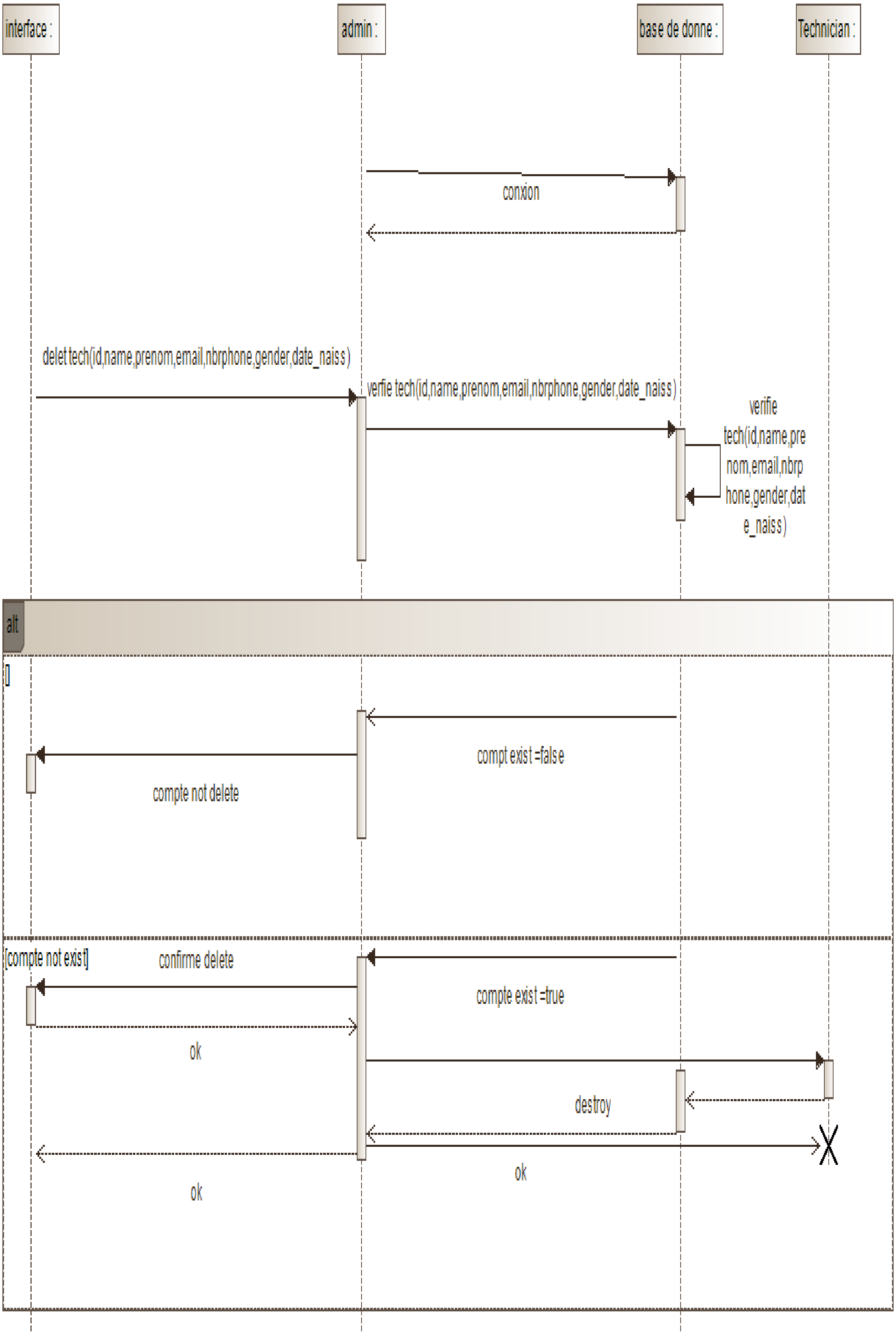

interface :
admin :
base de donne :
Technician :
conxion
delet tech(id,name,prenom,email,nbrphone,gender,date_naiss)
verfie tech(id,name,prenom,email,nbrphone,gender,date_naiss)
verifie tech(id,name,prenom,email,nbrphone,gender,date_naiss)
alt
[]
compt exist =false
compte not delete
[compte not exist]
confirme delete
compte exist =true
ok
destroy
ok
ok

Figure 2.9: Sequence diagrams of delete technician.

3.4.5: Sequence diagrams of all function technician: figure shows the diagram of all function technician account:

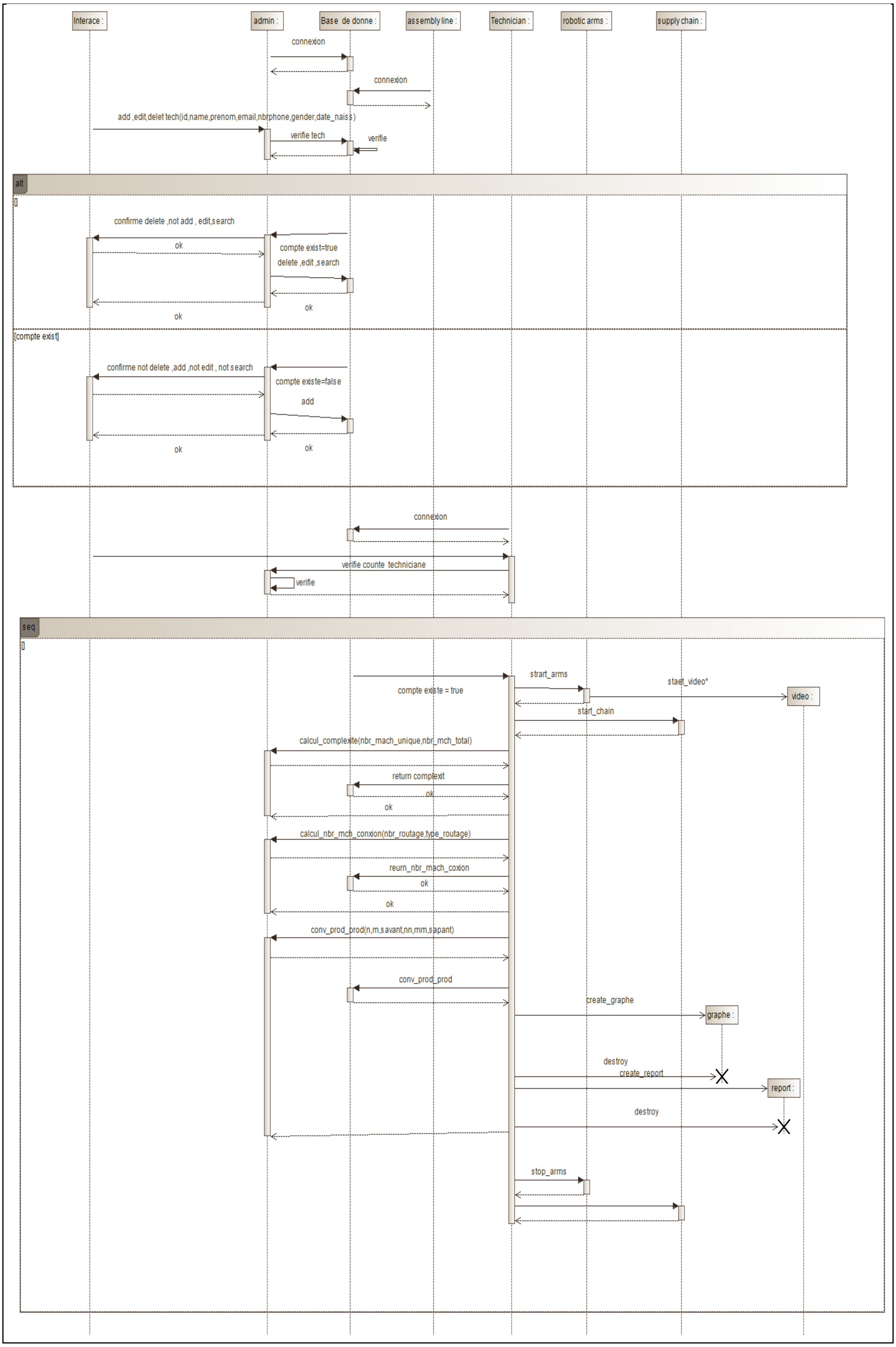


Figure 2.10: Sequence diagrams of all function of technician.

3.5 Activity Diagram:

In this section, we present the different activity diagrams of our system, which represents the interaction between the different related actors.

3.5.1-Robotic arm:

The diagram below (figure 2.10) represents the interaction between the robotic arms.

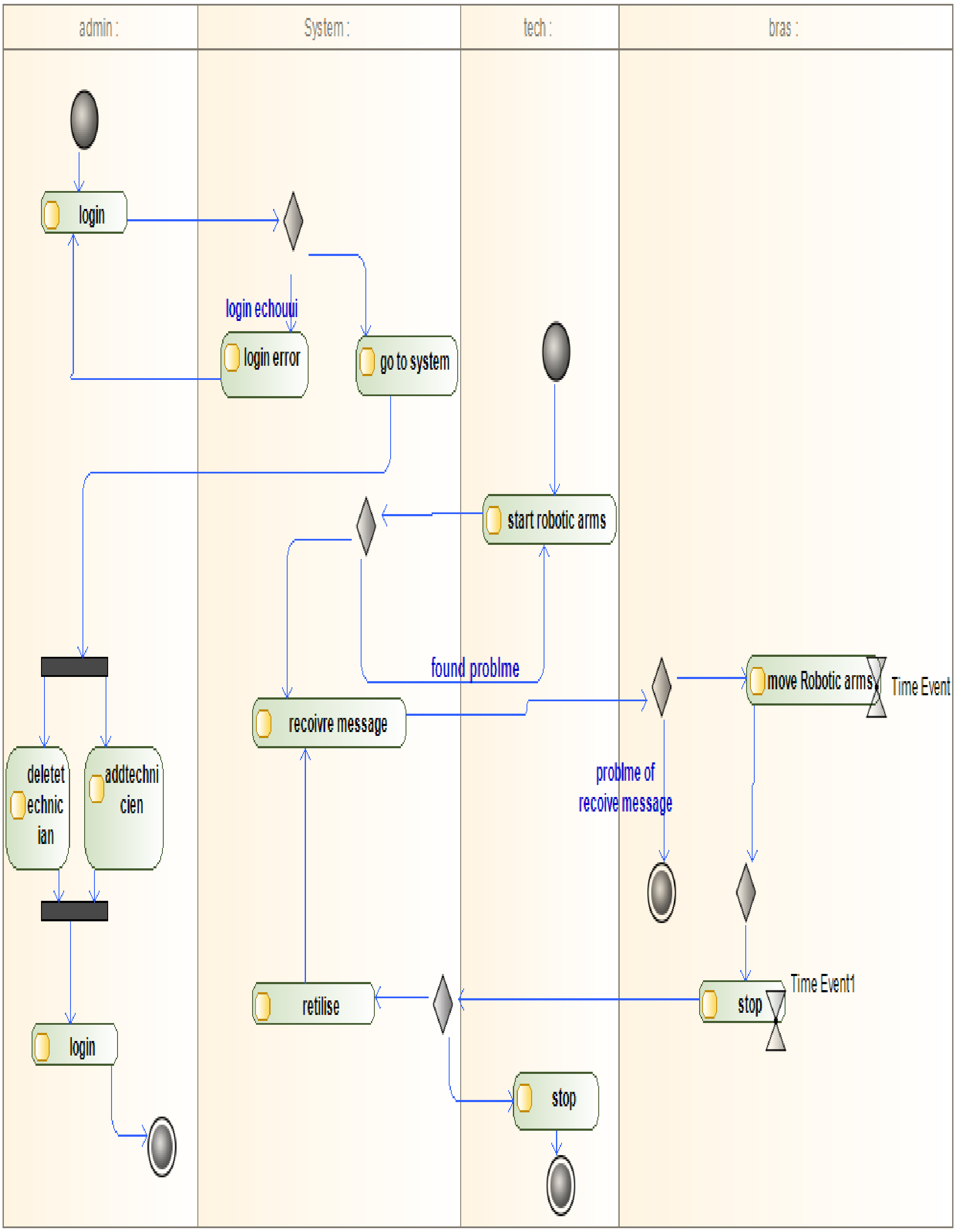


Figure 2.11: *Activity diagrams of Robotic arms.*

3.5.2-Assembly line:

The diagram below (figure 2.11) represents the interaction between the supply chains.

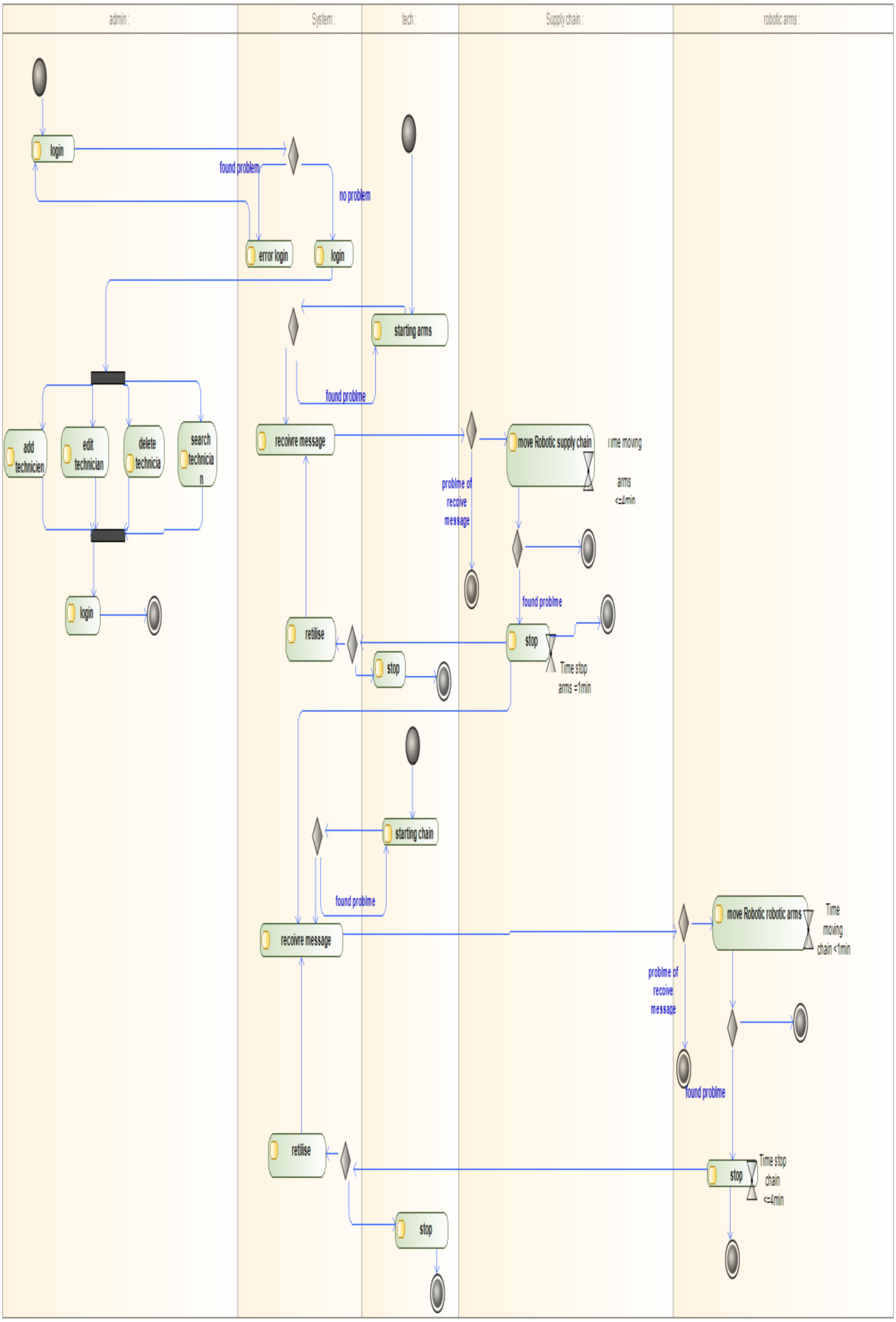


__Figure 2.12:__ Activity diagrams of supply chain.

4. Formal specifications business processes:

In this part, we will study the system specification by a system of transitions (timed automaton).

4.1- Formal specification software tools:

UPAAL [31]: is an integrated tool environment for modeling, validation and verification of real-time systems modeled as networks of time automatons, extended with data types (limited integers, arrays, ..). It was jointly developed by the universities of Uppsala (Sweden) and Aalborg (Denmark). It allows for the analysis of the network of timers communicating through binary synchronization and using broadcast / reception channels. Automata with added entire variables, clock tables, emergency,. . . Transitions manipulate two kinds of variables: clocks that evolve synchronously over time and discrete bounded variables. The state of the automaton may contain a condition on the clocks, called invariant, which must be satisfied by the time in this state. The passage of the automata is marked by:

-A guard, which expresses a condition on the values of the variables (true by default). This condition should generally be compatible with the invariant of the original state of the transition, and it must be satisfied to make the transition,

-A synchronization of the form ! Or ?, the lack of synchronization indicating the automaton's internal action.

-Reset some clocks and update certain variable s Whole.

4.2-Formally Description:

The system modeled in this study is the simplified system of an assembly line. This system includes the following three synchronized processes.

4.2.1-Assembly line: this system contains two systems, Robotic arms, and the supply chain will be synchronization between them.

4.2.2-Robotic arm: this system starting made from whet we need of irons plats and in other steps, the installation form of the car with irons plats form.

4.2.3-Supply chain: this system can change steps between (step to another step).

4.3-Declaration of variables and system assembly line by UPPAAL:

The Uppaal tool is made up of 3 main parts:

- A graphic editor where timed processes can be described,

- A graphic simulator where you can have a view of the behavior of the system.

- A checker that allows you to check the different properties.

The editor itself is made up of two parts:

- **Declaration:** Contains whole variables, clocks, channels of synchronization, and constants.

  Chan move_bras, pren_ves, tach_ves, rutilise, fin_bras, stop_bras, stop_chain, move_chain, fin_chain, remove_chain, remove_bras;
  Clock x;
  Bool mrc;

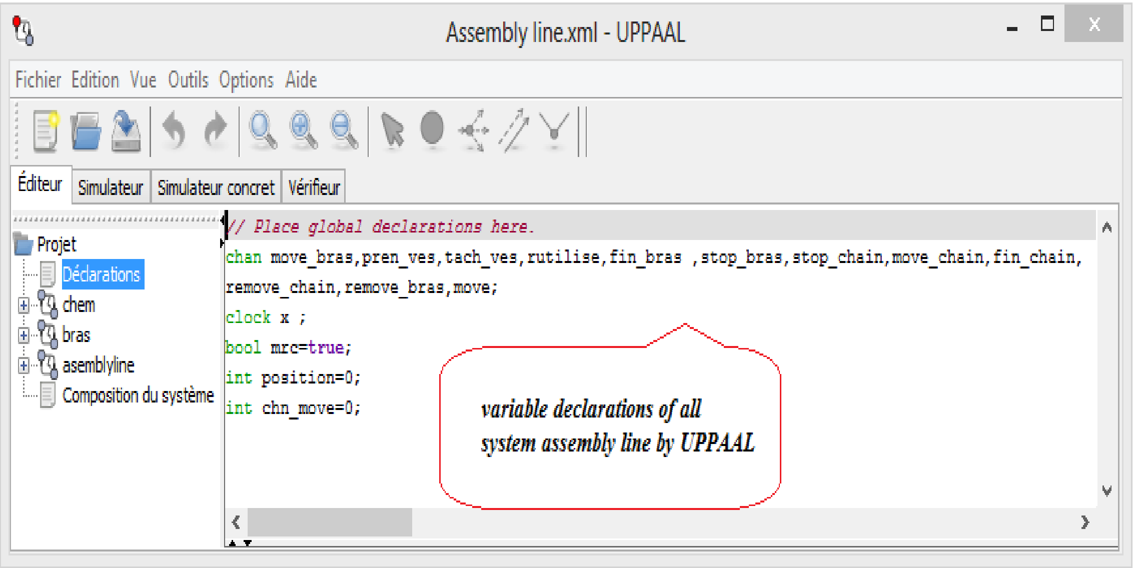
- **System declaration:** Contains processes.

```
// Place template instantiations here.
  brasd = bras();
 Chemad =chem ();
Asemblylin =asemblyline ();
```

// List one or more processes to be composed into a system.

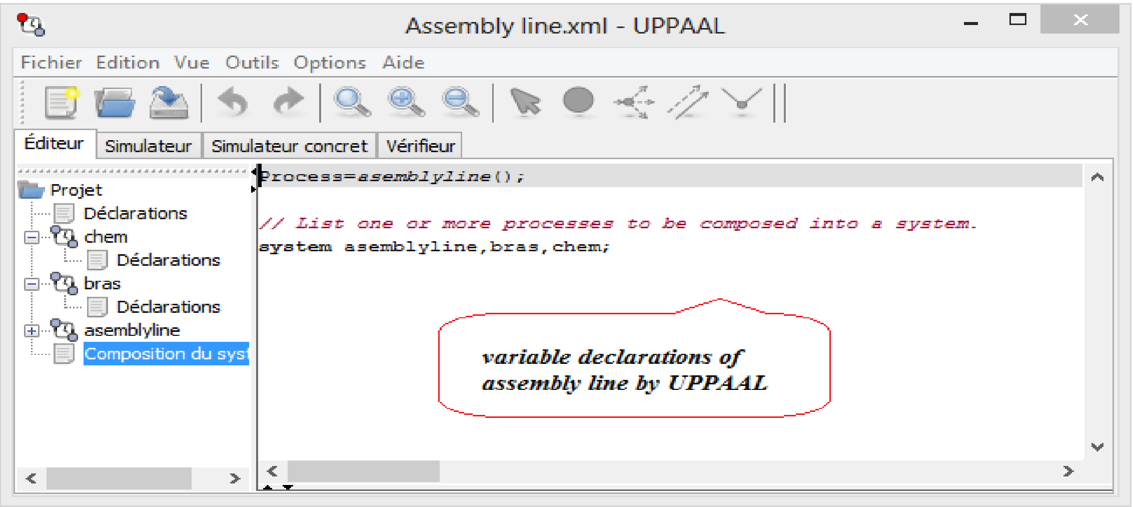
System asemblylin, brasd, chemad;

These statements are presented in Figure (2.13) variable declarations of the system assembly line and in Figure (2.14) sub_system declaration, which shows the Graphic User (GUI Interface) of UPPAAL.


Assembly line.xml - UPPAAL
Fichier Edition Vue Outils Options Aide
Éditeur Simulateur Simulateur concret Vérifieur
Projet
Déclarations
chem
bras
asemblyline
Composition du système
variable declarations of all system assembly line by UPPAAL

```
// Place global declarations here.
chan move_bras,pren_ves,tach_ves,rutilise,fin_bras ,stop_bras,stop_chain,move_chain,fin_chain,
remove_chain,remove_bras,move;
clock x ;
bool mrc=true;
int position=0;
int chn_move=0;
```

Figure (2.13): Variable declarations of assembly line by UPPAAL.


Assembly line.xml - UPPAAL
Fichier Edition Vue Outils Options Aide
Éditeur Simulateur Simulateur concret Vérifieur
Projet
Déclarations
chem
Déclarations
bras
Déclarations
asemblyline
Composition du syst
variable declarations of assembly line by UPPAAL

```
Process=asemblyline();

// List one or more processes to be composed into a system.
system asemblyline,bras,chem;
```

Figure (2.14): Sub_system of assembly line declaration by UPPAAL.

4.3.1 - Business Process Actions:

-move_bras: Starting move of robotic arms.

- pren_ves: Download the required installation tools.

-tach_ves: Install the required tools.

-Rutilise: Restarting move of Robotic arms.

-Fin_bras: fin moving of Robotic arms.

-Stop_bras: Stop moving of arm in short time

- Stop_Chain: Stop moving of chain in short time.

-move_chain: Starting move of chain.

-fin_ch: Fin moving of chain.

-remove_chain: Restarting move of chain.

-remove_bras: Restarting move of arm.

- chn_move: Counter moving of supply chain.

- position: Counter moving of robotic arm.

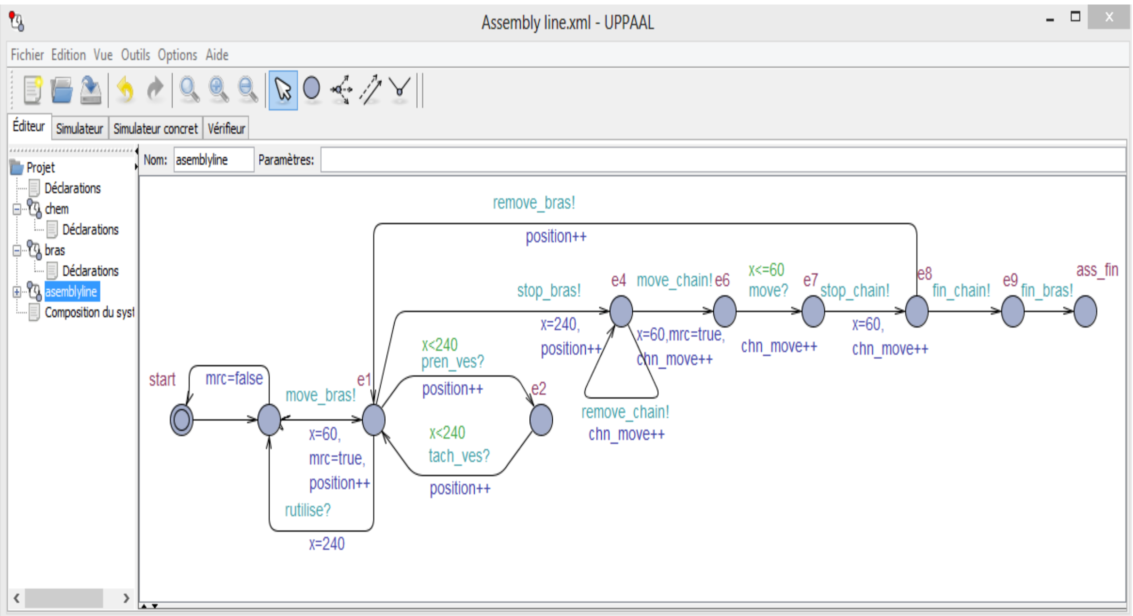
? : This operation is in sync with another, and this operator means that the subsystem has to wait for another sub-system to trigger the action.

! : This operator means that the action is done by this part of the system.

4.3.2- Formal Model Specification:

An assembly line is made up of three processes that synchronize with each other as follows: the assembly line, robotic arms and supply chain, they are modeled as state automatons finished in the following part:

- **_Business process_ Assembly line:** This process is the system of Assembly line, it is contains 10 states and it is synchronized with two other robotic arm and supply chain subsystem (figure 2.15):



Nom: asemblyline

Paramètres:

Figure (2.15): *Business process models of Assembly line.*

- **_Business process_ Robotic arms:** This process is the system of robotic arms, it is contains 6 states and it is synchronized with assembly line subsystem (figure 2.16):

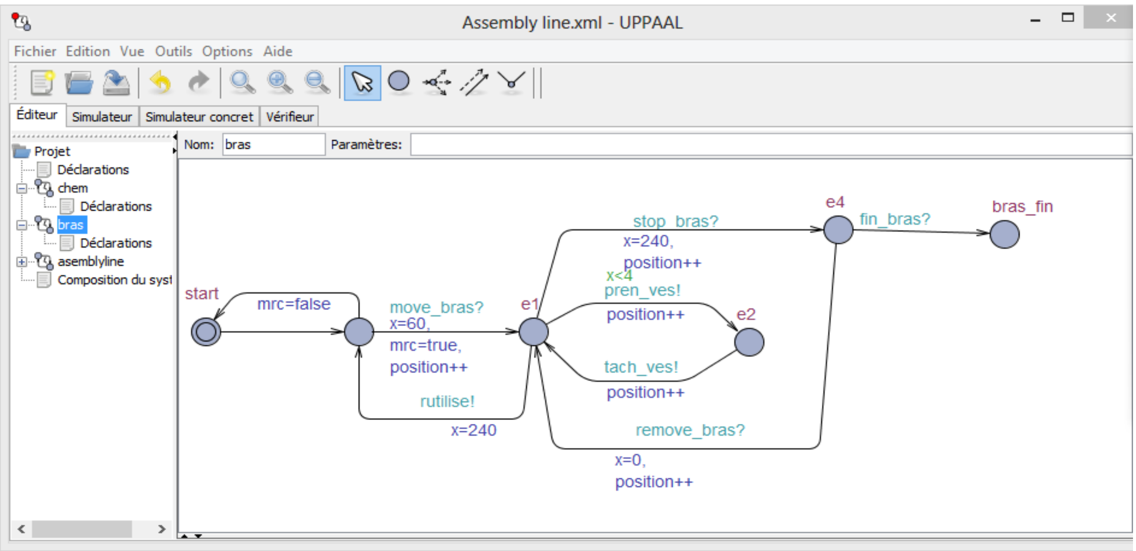


Figure (2.16): Business process models of Robotic arm.

- ***Business process* Supply chain:** This process is the system of, it is contains 6 states and it is synchronized with assembly line subsystem (figure 2.17):

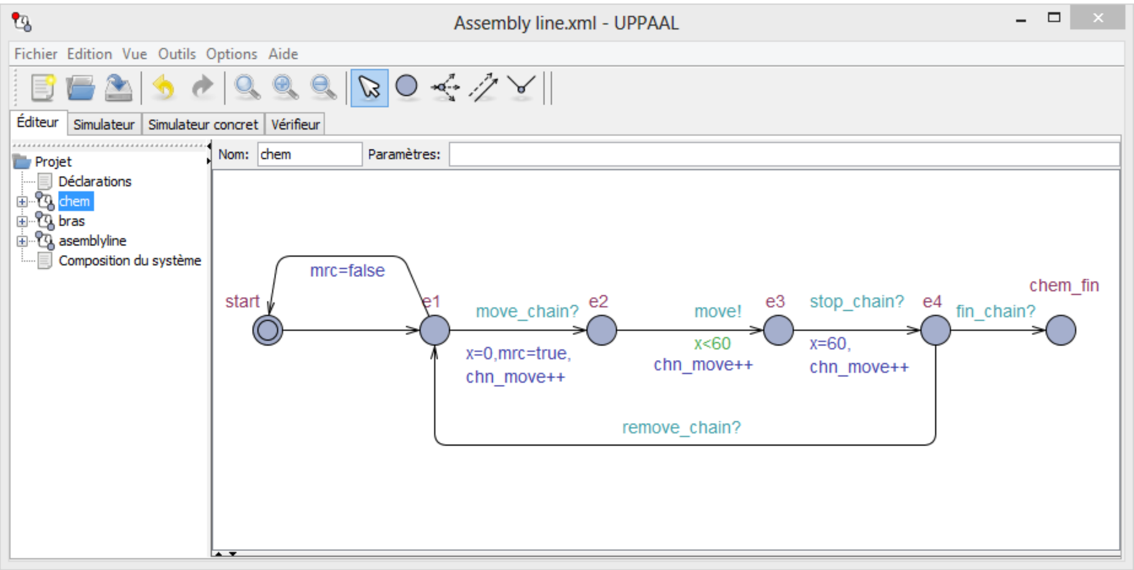


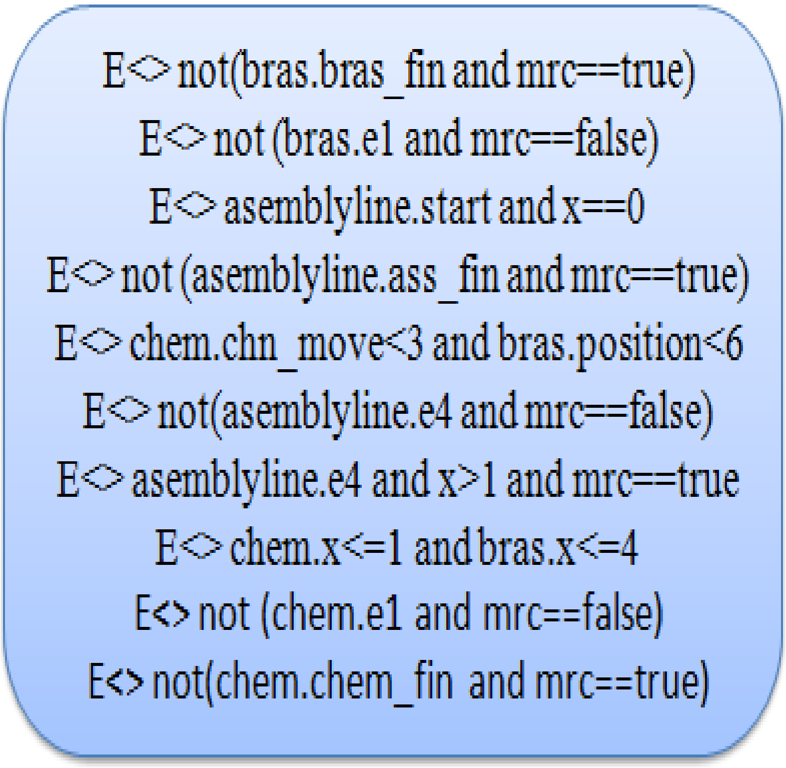
Figure (2.17): Business process models of supply chain.

- **Business *Process* synchronization:** three processes that synchronize the assembly line, robot arms and supply chain: For example, synchronization between "robot arms" and "supply chain" where "stop_bras" sends an action to and "delete_bras" "receives this action to check what has been done or not.
- **Business Process Guards:** Keepers express conditions regarding clock variables and variables that must be met. Formally, the keepers are a combination of time constraints and constraints on whole variables. For example in subsystem "chain" there is a guard (vair-true) between state e1 and e2.

- **Business Process Reset operation:** Res tying a clock or variable transition data is an initialization of the value of the clock. For example the under «bras" system in the transition between (s1 s2) after the 4 min the (in the "chain" process) the was reset to zero to calculate the time it was closed bras valve for up to restart the system.

4.2.3- Verification of Formal System Modeled:

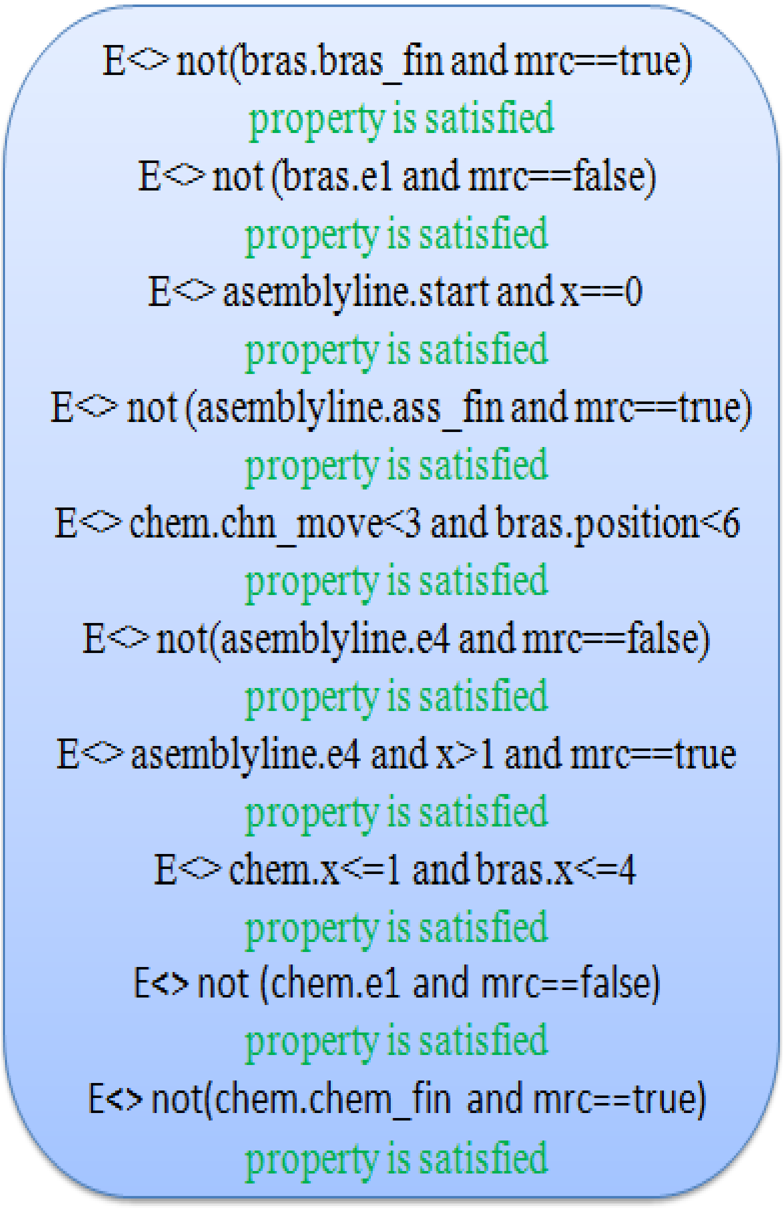
The purpose of verification is to ensure that a program meets many characteristics. Model-checking is an automatic formal verification technique, for which it is necessary to formally model the behavior of the system. In addition, the temporal logic CTL has also been well presented; it is an attribute specification language. Once the system is described by the conversion system and the required attributes are specified in the time logic, an algorithm called model checking will automatically answer the question: "Does the system meet the required attributes? ». We have written the formal the model standardizes a pipeline to the idea CTL time logic as shown in the Figure **(2.18**).



```
E<> not(bras.bras_fin and mrc==true)
E<> not (bras.e1 and mrc==false)
E<> asemblyline.start and x==0
E<> not (asemblyline.ass_fin and mrc==true)
E<> chem.chn_move<3 and bras.position<6
E<> not(asemblyline.e4 and mrc==false)
E<> asemblyline.e4 and x>1 and mrc==true
E<> chem.x<=1 and bras.x<=4
E<> not (chem.e1 and mrc==false)
E<> not(chem.chem_fin and mrc==true)
```

Figure (2.18): *Formal specification of assembly line CTL.*

Using the Model-Checking algorithm as a verification technique formal to prove the safety of our formal specification under temporal logic Assembly line system CTL, Figure (2.19) shows the satisfaction of the properties Requested.



E<> not(bras.bras_fin and mrc==true)

property is satisfied

E<> not (bras.e1 and mrc==false)

property is satisfied

E<> asemblyline.start and x==0

property is satisfied

E<> not (asemblyline.ass_fin and mrc==true)

property is satisfied

E<> chem.chn_move<3 and bras.position<6

property is satisfied

E<> not(asemblyline.e4 and mrc==false)

property is satisfied

E<> asemblyline.e4 and x>1 and mrc==true

property is satisfied

E<> chem.x<=1 and bras.x<=4

property is satisfied

E<> not (chem.e1 and mrc==false)

property is satisfied

E<> not(chem.chem_fin and mrc==true)

property is satisfied

Figure (2.19): *Verification result of each specification.*

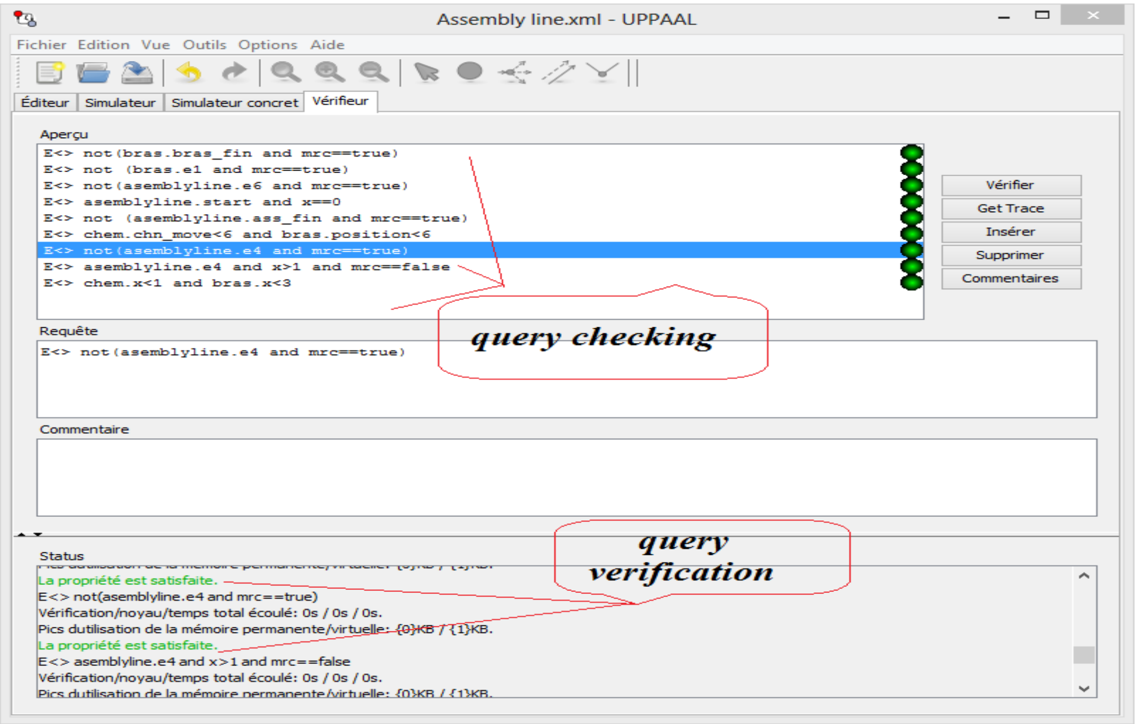
The following table summarizes the formal and informal specification of an assembly line (Table 2.3):

Formally Description	**Properties Formal**	**Result**
T1: moving of supply chain duration time =1 min (60s) and moving robotic arms duration time= 4 min (240s).	E<> chem.x<=1 and bras.x<=4	property is satisfied

M1: For re moved supply chain, message received exit and time >1min.	E<> asemblyline.e4 and x>1 and mrc==true	property is satisfied
S1: The message received not found, the supply chain not moving. S2: The assembly line stop and the message received not apply. S3: Assemly line time starting move x=0.	E<> not(asemblyline.e4 and mrc==false) E<> not (asemblyline.ass_fin and mrc==true) E<> asemblyline.start and x==0	property is satisfied
Ps1: Moves number of supply chain <3 and moves number of robotic arms <6.	E<> chem.chn_move<3 and bras.position<6	property is satisfied
C1: The supply chain not start and the message received not apply. C2: The supply chain stop and the message received not apply.	E<> not (chem.e1 and mrc==false) E<> not(chem.chem_fin and mrc==true)	property is satisfied
B1: The robotic arms not start and the message received not apply. B2: The robotic arms stop and the message received not apply.	E<> not (bras.e1 and mrc==false) E<> not(bras.bras_fin and mrc==true)	property is satisfied

Tableau (2.3): *Detailed representation of a formal specification.*

In conclusion, the following figure (Figure 2.20) explains the result of verification with UPPAAL.


Assembly line.xml - UPPAAL
Fichier Edition Vue Outils Options Aide
Éditeur Simulateur Simulateur concret Vérifieur
Vérifier
Get Trace
Insérer
Supprimer
Commentaires
query checking
query verification

Aperçu

```
E<> not(bras.bras_fin and mrc==true)
E<> not (bras.e1 and mrc==true)
E<> not(asemblyline.e6 and mrc==true)
E<> asemblyline.start and x==0
E<> not (asemblyline.ass_fin and mrc==true)
E<> chem.chn_move<6 and bras.position<6
E<> not(asemblyline.e4 and mrc==true)
E<> asemblyline.e4 and x>1 and mrc==false
E<> chem.x<1 and bras.x<3
```

Requête

```
E<> not(asemblyline.e4 and mrc==true)
```

Commentaire

Status

```
La propriété est satisfaite.
E<> not(asemblyline.e4 and mrc==true)
Vérification/noyau/temps total écoulé: 0s / 0s / 0s.
Pics dutilisation de la mémoire permanente/virtuelle: {0}KB / {1}KB.
La propriété est satisfaite.
E<> asemblyline.e4 and x>1 and mrc==false
Vérification/noyau/temps total écoulé: 0s / 0s / 0s.
Pics dutilisation de la mémoire permanente/virtuelle: {0}KB / {1}KB.
```

Figure (2.20): *Result of formal verification by UPPAAL.*

4.3.4- validation of Formal system Modeled by the simulation:

In this part, we used the UPPAAL simulator to validate the behavior of our system in order to show if there are problems such as: loops infinite, blocking ... etc. We have encountered problems like this, the following figure showing the validation of our system by the simulation of Uppaal (figure 2.21).

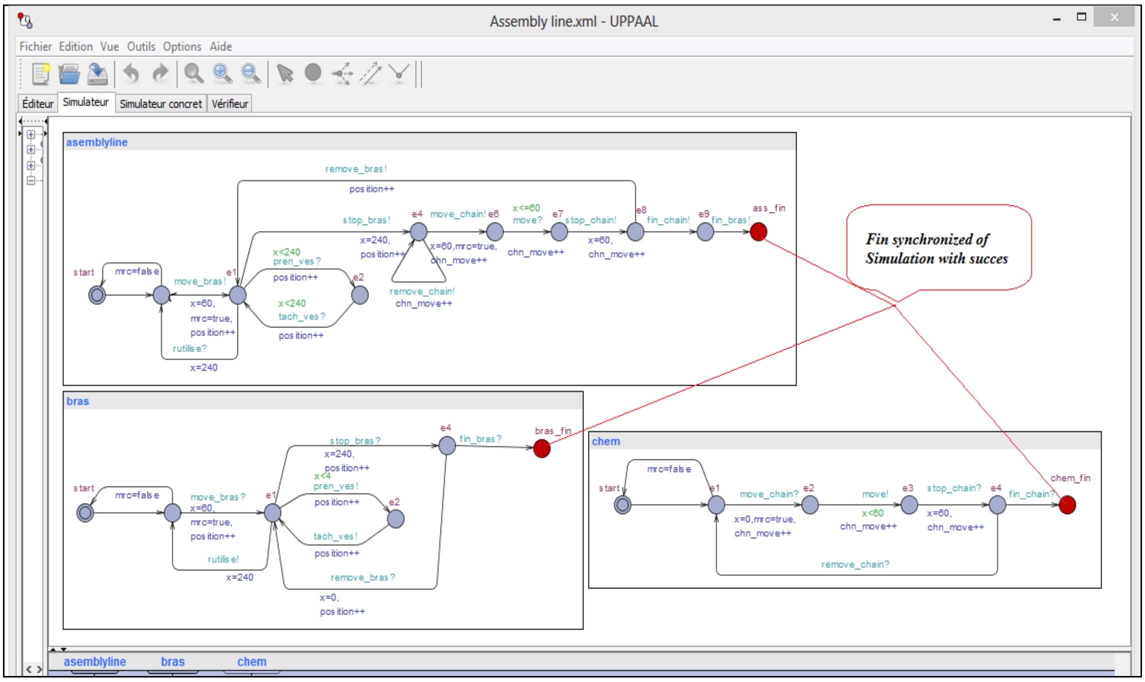
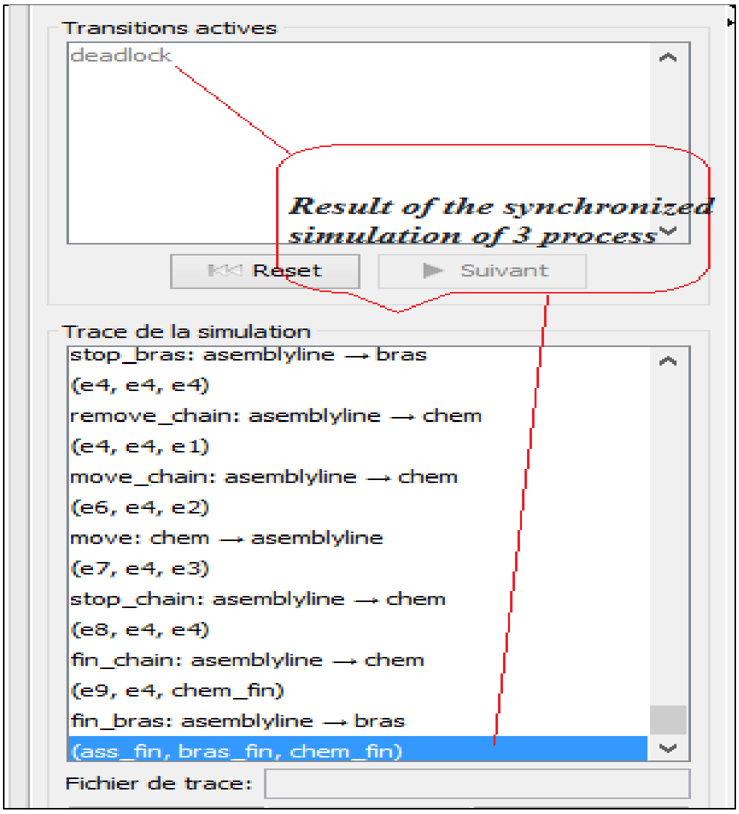


Figure (2.21): *Simulation of synchronized behavior.*

After the passage of the different synchronized processes, the UPPAAL shows the result deterministic following (figure 2.22):



```
deadlock
```

Result of the synchronized simulation of 3 process

```
stop_bras: asemblyline → bras
(e4, e4, e4)
remove_chain: asemblyline → chem
(e4, e4, e1)
move_chain: asemblyline → chem
(e6, e4, e2)
move: chem → asemblyline
(e7, e4, e3)
stop_chain: asemblyline → chem
(e8, e4, e4)
fin_chain: asemblyline → chem
(e9, e4, chem_fin)
fin_bras: asemblyline → bras
(ass_fin, bras_fin, chem_fin)
```

Figure (2.22): *Result of the synchronized simulation.*

4.3.5-Analysis

The formal method involves the application of mathematical techniques to design and implement software [40]. A formal specification is expressed in a language whose syntax and semantics are formally defined. Model-based specification uses set of theory, function and logic tools to develop abstract models of the system**.**

As well as the formal specification of a critic systems provides an important benefits:

Advantages:

-Discover ambiguities, incompleteness and inconsistencies in the software.

-Provide defect-free software.

-After each iteration, the incremental growth of effective solutions.

-The model does not involve high complexity.

-The semantic verification of formal specification language is self-consistent.

However the formal specification of a critic systems surf from some problems:

Disadvantages:

-Time-consuming is expensive.

-It is difficult to use this model as a communication mechanism for non-technical personnel.

-A lot of training is required because only a few developers have the basic knowledge to implement this model.

The following table presents our proper comparison between formally, semi_formally, and informally paradigms (Table 2.4):

	Temps deliberates	**Verification/ Validation**	**Refinement**	**Graphique**	**application Fields**	**User catégorie**
Formally	Weak (specification content a lot of tapes)	Yes (true or false)	Weak (used complexes)	No (expressions mathemaique)	Specification, Complex or critic systems	Good Conceptor, Mathemecian, best testor
Semi-Formally	Average (analyse systems taks a time)	No (not pas exist expressions mathematique)	Average (before used Need analyse system)	Yes (based on graphique)	Analysis, simple systems	Analyste
Informally	Fast (description fast operation)	No (not pas exist expressions mathematique)	Fast (used easy)	No (dased on logique)	Systems description	Costumer

Table (2.4): *Comparison between paradigms.*

5. CONCLUSION

In this chapter, we selected assembly line cars as a study case, we introduced his system components, define them and model their business processes using UML.

In the second part of our study, we used UPPAAL for specification and formal verification of our system, elaborate their automaton time synchronal, define all walk probabilities, after that validating by CTL if the automaton time true or false. As result although it a long time to study but, it gives us a lot of positives. We definite all the action and the ingredient what will want, knowing the probability source of risk robotic and the safety system for the safety of its stages.

In the next chapter, we will talk about the implementation side of our system simulation which includes the software tools used in the development phase, then we will present the different scenarios of our simulation by showing multiple graphical user interfaces (GUI).

CHAPITRE 03 :

IMPLEMENTATION

CHAPTER 3: IMPLEMENTATION

1-INTRODUCTION

This chapter is dedicated to the implementation of the code by using information from the modeling, we will treat the different steps of code generation of the code, and interested in the passage according to the specificities of the various type of semi-formal and formal models already developed towards programming, and over time, we have proposed recommendations for the use of any formalism.

We will start with the description of the hardware and software environment and software environment while giving an overview of the work done during preprogramming period.

2- DEVELOPMENT TOOLS, ENVIRONMENTS AND LANGUAGES

In this section, we will introduce the necessary tools, software and programming languages used in our development process application and our use case we have chosen the following software tools:

2.1- Programing language

2.1.1- JAVA FX [59]: Is an Oracle software development platform designed to promote desktop and rich Internet applications (RIA) that can be accessed from a variety of devices. The platform was originally developed by Sun Microsystems and consists of Java FX Script and Java FX Mobile. The scripting language is designed to make it easier to use the Java Swing user interface library to create rich user interfaces that will run anywhere that supports the Java Standard Edition. Java FX supports the creation of GUI in a declarative language format, which defines the interface components and various display effects in Java's Swing interface toolkit, including animation, vector graphics, audio tracks and videos, all of which are in the Java virtual machine (JVM). (The name is a short way to type "Java effects.") **.**

2.1.2-Cascading Style Sheets (CSS) [61]: Is a language that allows you to manage the presentation of a Web page. The CSS language is a recommendation of the World Wide Web Consortium, just like HTML or XML.

The styles allow you to define rules applied to one or more HTML documents. Indeed, with HTML, you can define both the structure and the presentation. But this poses some problems. With the couple HTML/CSS, we can create web pages where the structure of the document is in the HTML file while the presentation is in a CSS file.

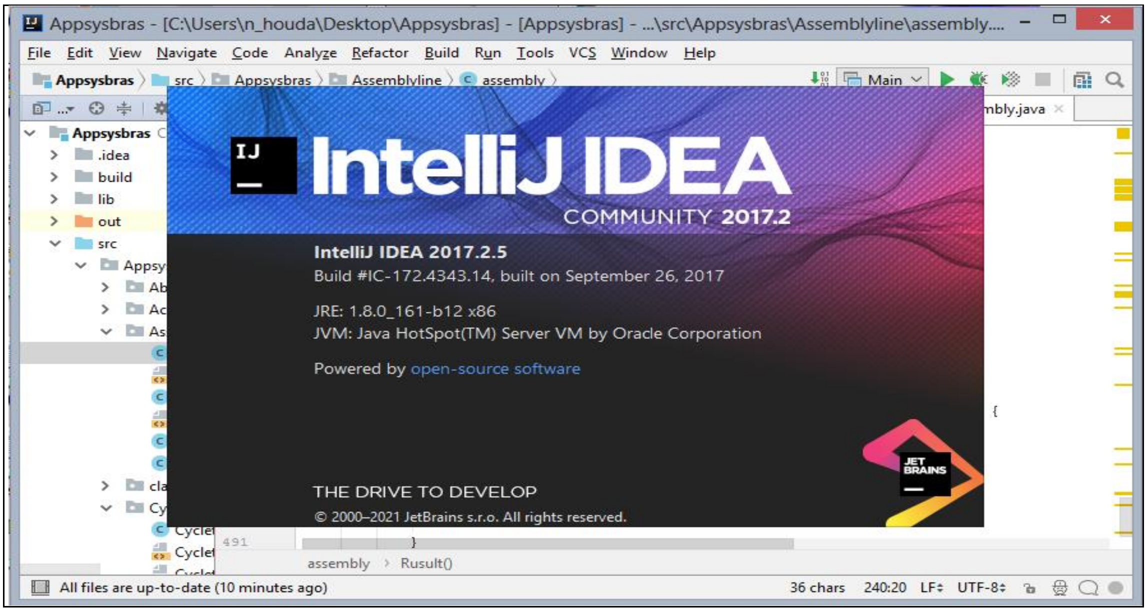
CSS also allows the definition of different rules for each display support. CSS also allows improving the accessibility of web documents.

Moreover, CSS adds new functionalities compared to HTML from a style point of view.

2.2- Programing environment

2.2.1-IntelliJ IDEA [62]: is a special programming environment or integrated development environment (IDE) mainly used for Java. This environment is dedicated to program development. It was developed by a company called JetBrains, whose official name is IntelliJ. It has two versions: a community version authorized by Apache 2.0 and a commercial version called the ultimate version. They can all be used to create software that can be sold. IntelliJ IDEA is different from similar products in its ease of use, flexibility and reliable design.

The figure below (Figure 3.1) represents Intellij IDEA.



Appsysbras - [C:\Users\n_houda\Desktop\Appsysbras] - [Appsysbras] - ...\src\Appsysbras\Assemblyline\assembly....

IntelliJ IDEA
COMMUNITY 2017.2

IntelliJ IDEA 2017.2.5
Build #IC-172.4343.14, built on September 26, 2017

JRE: 1.8.0_161-b12 x86
JVM: Java HotSpot(TM) Server VM by Oracle Corporation

Powered by open-source software

THE DRIVE TO DEVELOP
© 2000–2021 JetBrains s.r.o. All rights reserved.

All files are up-to-date (10 minutes ago)

Figure (3.1) : Intellij IDEA.

2.2.2-Screne builder [60]: is a visual layout tool that allows users to quickly design the user interface of Java FX applications without coding. Users can drag and drop user interface components to the workspace, modify their properties, apply style sheets, and the FXML code of the layout being created will be automatically generated in the background. The result is an FXML file, which can then be linked to a Java project by binding the user interface to the application logic. :

• Views are FXML files that describe the user interface.

• The controller is a Java class, and you can choose to implement an initialization class, which is declared as a controller in the FXML file.

• The model is composed of domain objects, defined on the Java side, and can be connected to the view through the controller.

The figure below (Figure 3.2) represents screne builder.

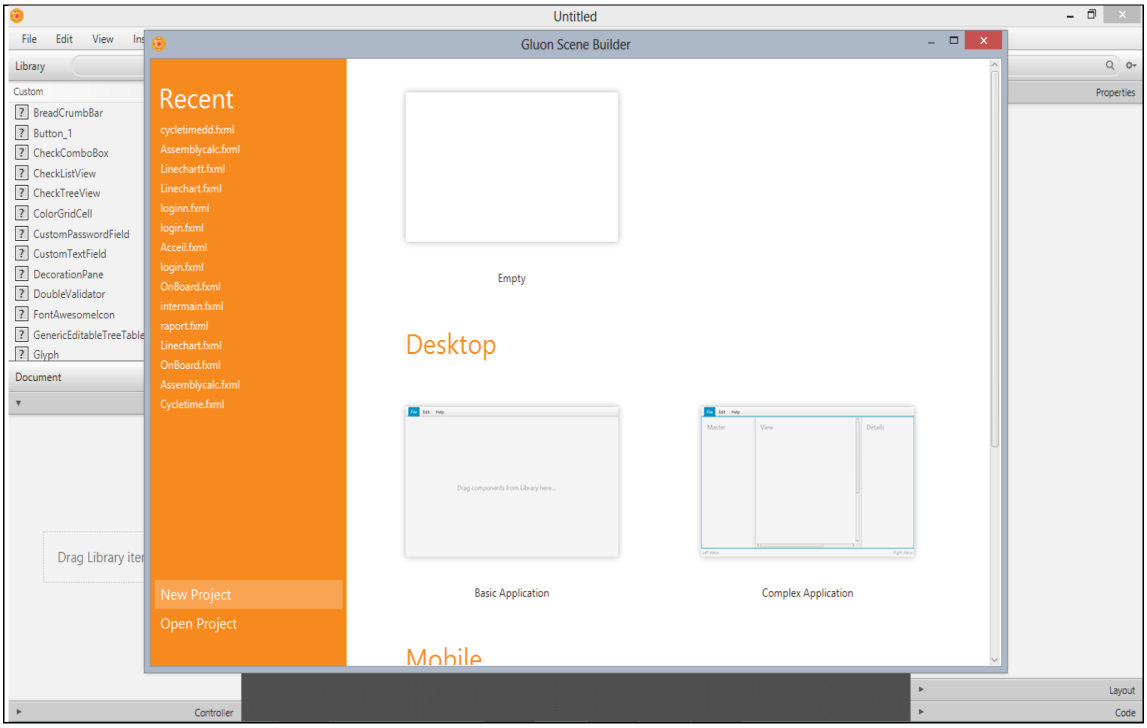


Figure (3.2): screne builder.

2.2.3-TIBCO-jaspersoft studio [49]: completely rewrites iReport Designer and can be used as an Eclipse plug-in and a standalone application. Jaspersoft Studio allows you to create complex layouts including charts, images, sub-reports, cross-tabs, etc. You can access your data through JDBC, TableModels, JavaBeans, XML, Hibernate, CSV and custom sources, and then publish your report as PDF, RTF, XML, XLS, CSV, HTML, XHTML, and text, DOCX or OpenOffice.

The figure below (Figure 3.3) represents TIBCO-jaspersoft studio.

Figure (3.3): TIBCO –jaspersoft studoi.

2.3 Database of application system:

In this section, we will introduce the database system that manages all our assembly lines according to classes the data following: admin, assembly line, configuration, cycle time, human, reports.

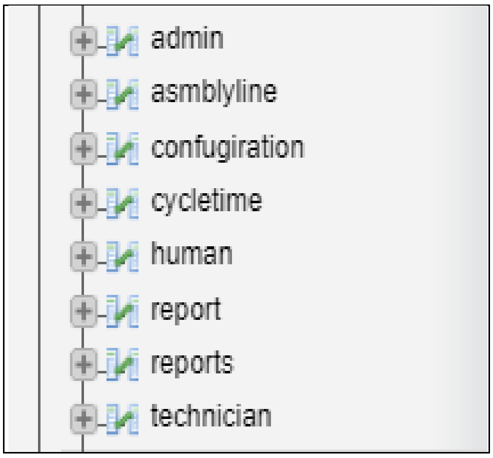
2.3.1- XAMPP [61]: is a set of software used to easily set up a Web server, an FTP server and an e-mail server. It is a free software distribution (X Apache MySQL Perl PHP) offering a good flexibility of use, recognized for its simple and fast installation.

Thus, it is within the reach of most people as it does not require specific knowledge and works, moreover, on the most common operating devices.

It is distributed with various software libraries that significantly expand the range of services: OpenSSL, Expat (XML parser), PNG, SQLite, zlib, ... but also various Perl and Tomcat modules. Many people criticize the quantity of extensions added which are for the most part useless for beginners. A lite version has therefore been put in place.

2.3.2 –Tables database:

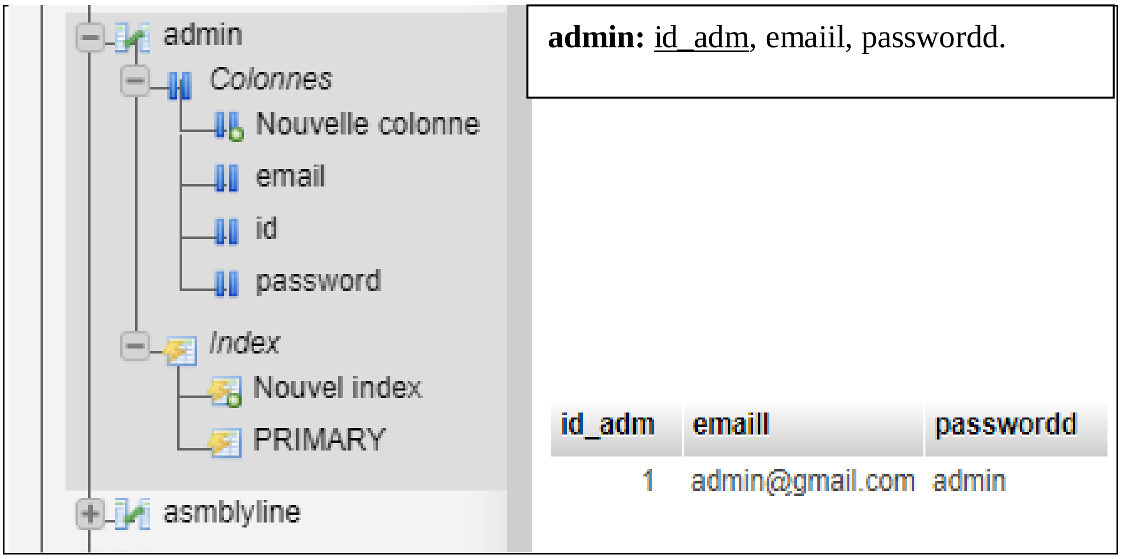
Xampp table's database of system application, the figure below (Figure 3.4) represents this table:



- admin
- asmblyline
- confugiration
- cycletime
- human
- report
- reports
- technician

Figure (3.4): database table.

1-Admin: represents the information of each admin and it contains the following: id, email, and password. The figure below (Figure3.5) represents that:



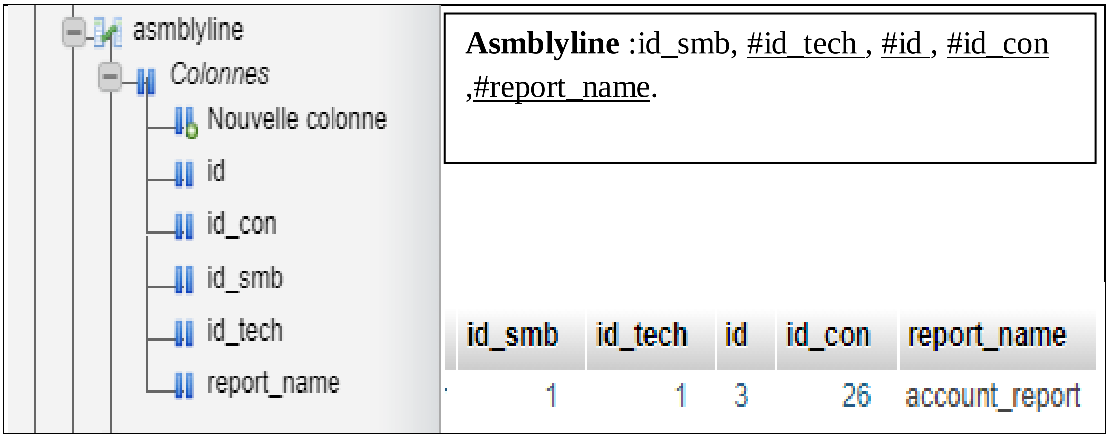
- admin
  - Colonnes
    - Nouvelle colonne
    - email
    - id
    - password
  - Index
    - Nouvel index
    - PRIMARY
- asmblyline

admin: id_adm, emaiil, passwordd.

id_adm	emaill	passwordd
1	admin@gmail.com	admin

Figure (3.5): table admin.

2-Asemblyline: represents the information of assembly line and it contains the following: buffer, id, PTm , TTm, and report_name. The figure below (Figure 3.6) represents that:

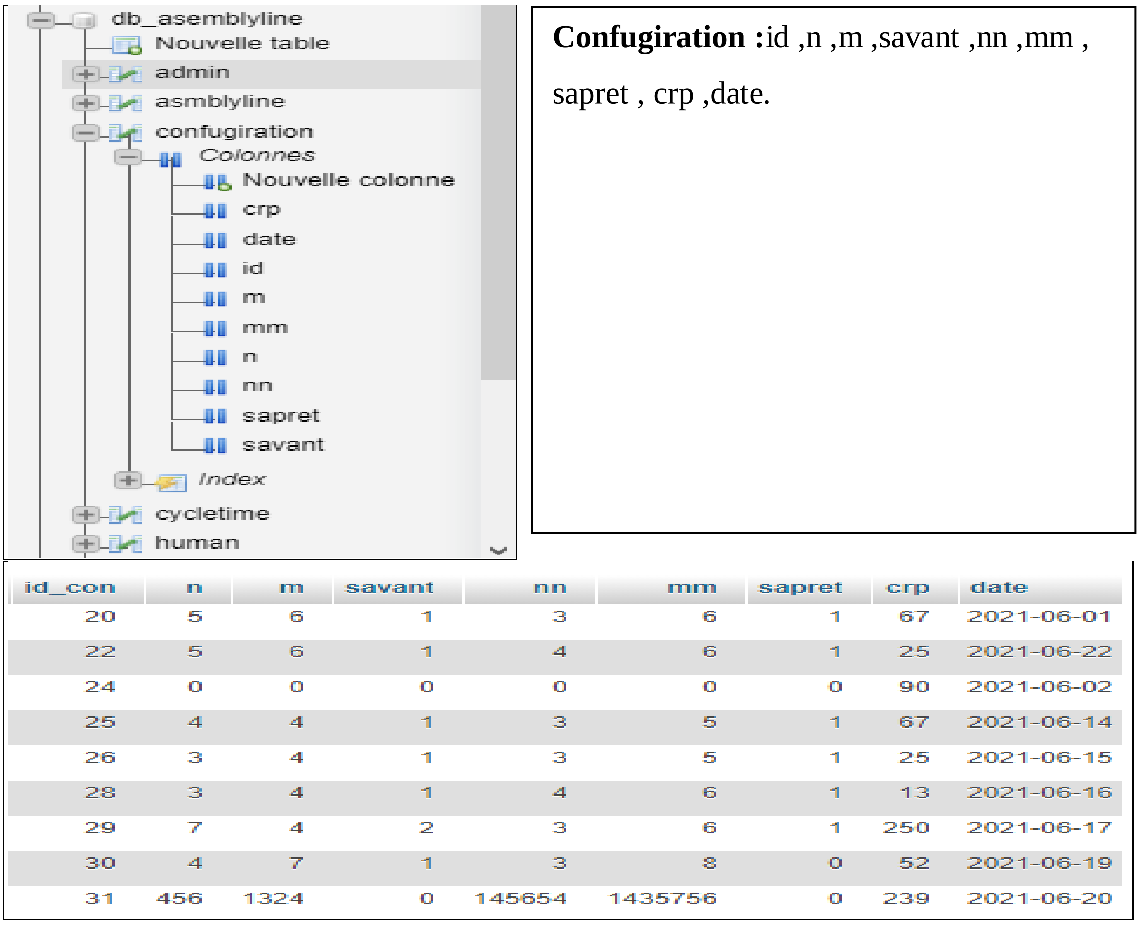


Asmblyline :id_smb, <u>#id_tech </u>, <u>#id </u>, <u>#id_con</u> ,<u>#report_name</u>.

id_smb	id_tech	id	id_con	report_name
1	1	3	26	account_report

Figure (3.6): table asemblyline.

2-Confugiration: represents the information of assembly line and it contains the following: crp, date, id, m, mm, n, nn, sapret, and savant. The figure below (Figure 3.7) represents that:

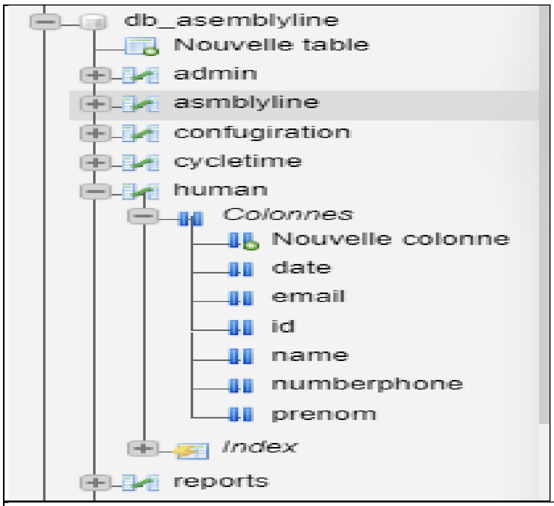


Confugiration :id ,n ,m ,savant ,nn ,mm , sapret , crp ,date.

id_con	n	m	savant	nn	mm	sapret	crp	date
20	5	6	1	3	6	1	67	2021-06-01
22	5	6	1	4	6	1	25	2021-06-22
24	0	0	0	0	0	0	90	2021-06-02
25	4	4	1	3	5	1	67	2021-06-14
26	3	4	1	3	5	1	25	2021-06-15
28	3	4	1	4	6	1	13	2021-06-16
29	7	4	2	3	6	1	250	2021-06-17
30	4	7	1	3	8	0	52	2021-06-19
31	456	1324	0	145654	1435756	0	239	2021-06-20

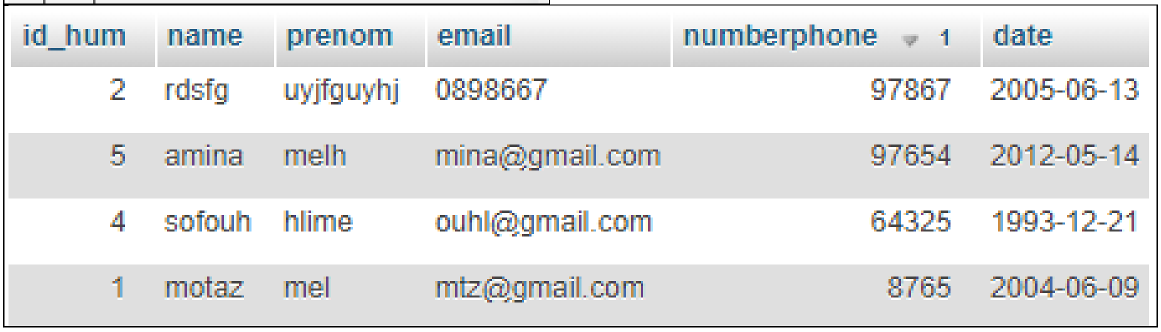
Figure (3.7): table confuguration.

3-Human: represents the information of human and it contains the following: date, email, id, numberphone, name, prenom. The figure below (Figure 3.8) represents that:



```
db_asemblyline
    Nouvelle table
    admin
    asmblyline
    confugiration
    cycletime
    human
        Colonnes
            Nouvelle colonne
            date
            email
            id
            name
            numberphone
            prenom
        Index
    reports
```

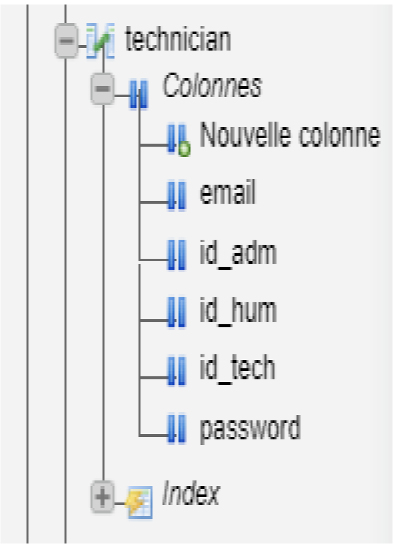
Human: id_hum, name, prenom , email, numberphone , date.



id_hum	name	prenom	email	numberphone ▼ 1	date
2	rdsfg	uyjfguyhj	0898667	97867	2005-06-13
5	amina	melh	mina@gmail.com	97654	2012-05-14
4	sofouh	hlime	ouhl@gmail.com	64325	1993-12-21
1	motaz	mel	mtz@gmail.com	8765	2004-06-09

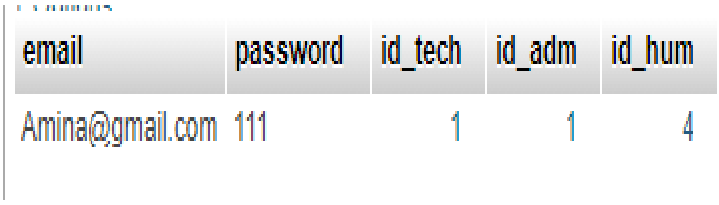
Figure (3.8): table human.

4-Technician: represents the information of technician and it contains the following: email, id_adm, id_hum, id_tech, password. The figure below (Figure 3.9) represents that:



```
technician
    Colonnes
        Nouvelle colonne
        email
        id_adm
        id_hum
        id_tech
        password
    Index
```

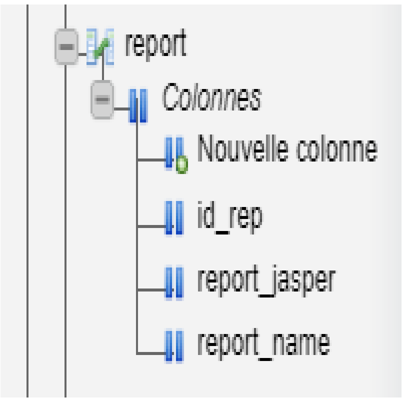
Technician:email, #id_adm,# id_hum, id_tech, password.



email	password	id_tech	id_adm	id_hum
Amina@gmail.com	111	1	1	4

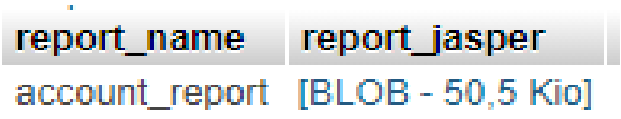
Figure (3.9): table technician.

5-Report: represents the information of technician and it contains the following: email, id_adm, id_hum, id_tech, password. The figure below (Figure 3.10) represents that:



- report
  - Colonnes
    - Nouvelle colonne
    - id_rep
    - report_jasper
    - report_name

Report :report_name,report_jasper.



report_name	report_jasper
account_report	[BLOB - 50,5 Kio]

Figure3.10: table report.

3-General diagram of application

The following diagram (Figure 3.11) explains the architecture of our application:

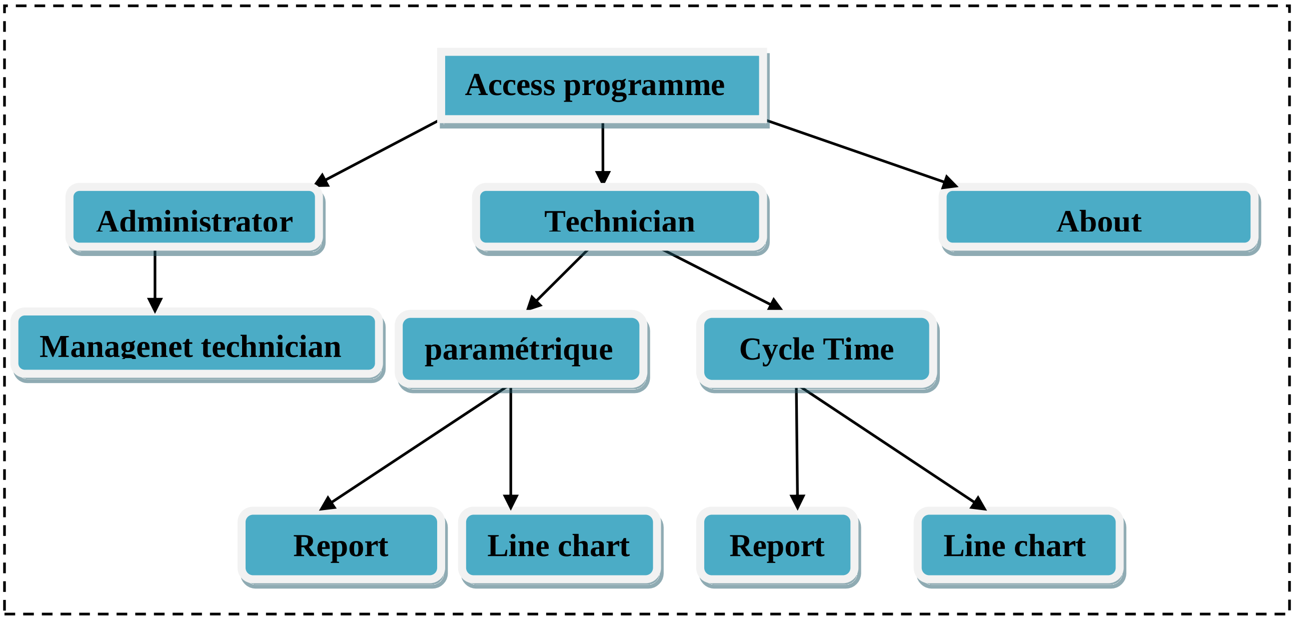


Figure (3.11): General diagram of application.

4-Presenting the developed applications

In this section, we present our application by providing GUIs and explaining how it functions.

4.1-The access interface:

The figure below (Figure3.12) presents the access interface:

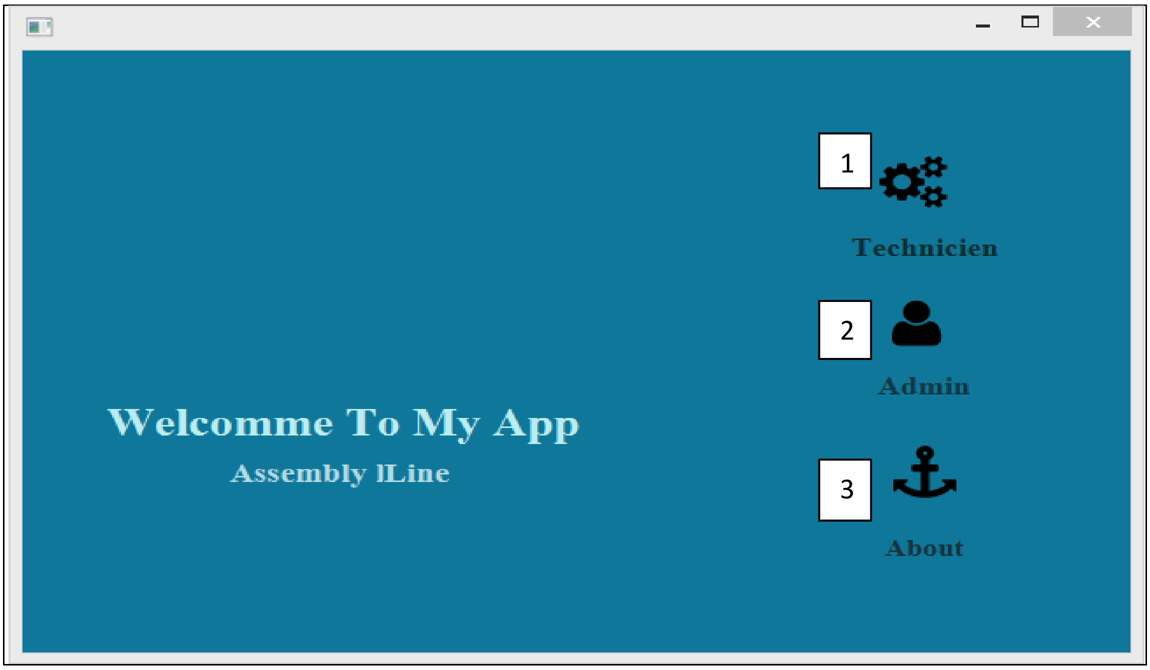


Figure (3.12): The access interface.

1- Form click go to the authentication technician interface.

2-Form click go to the authentication admin interface

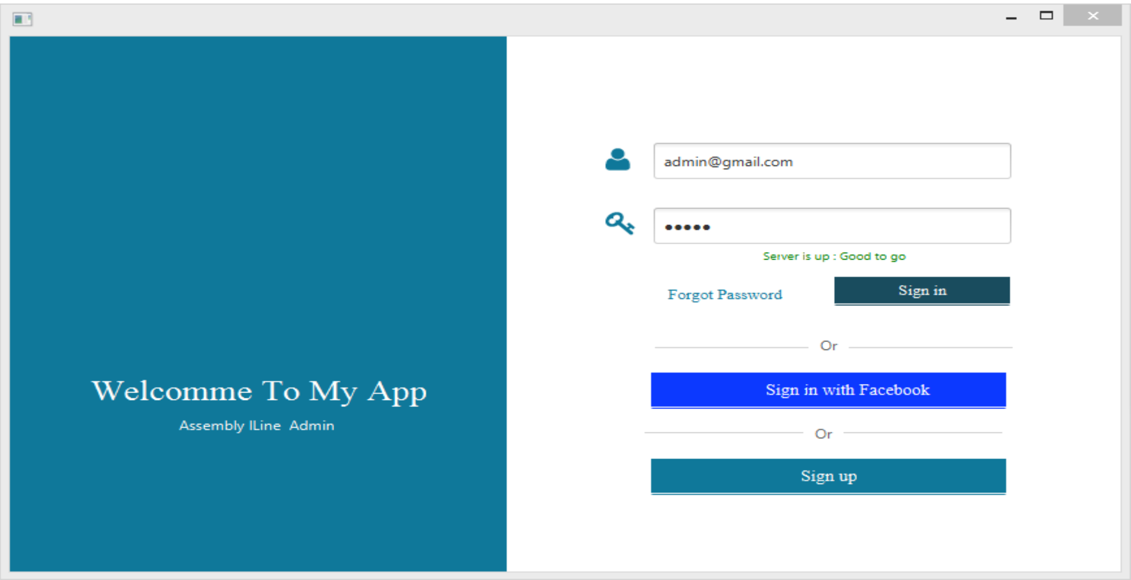
3-Form click go to the about interface.

4.2-The authentication interface:

They are two interface authentication interfaces (admin and technician).

A- The authentication admin interface

The figure below (Figure3.13) presents the authentication admin interface:



admin@gmail.com

•••••

Server is up : Good to go

Figure (3.13): the authentication admin interface.

B- The authentication technician interface

The figure below (Figure3.14) presents the authentication technician interface:

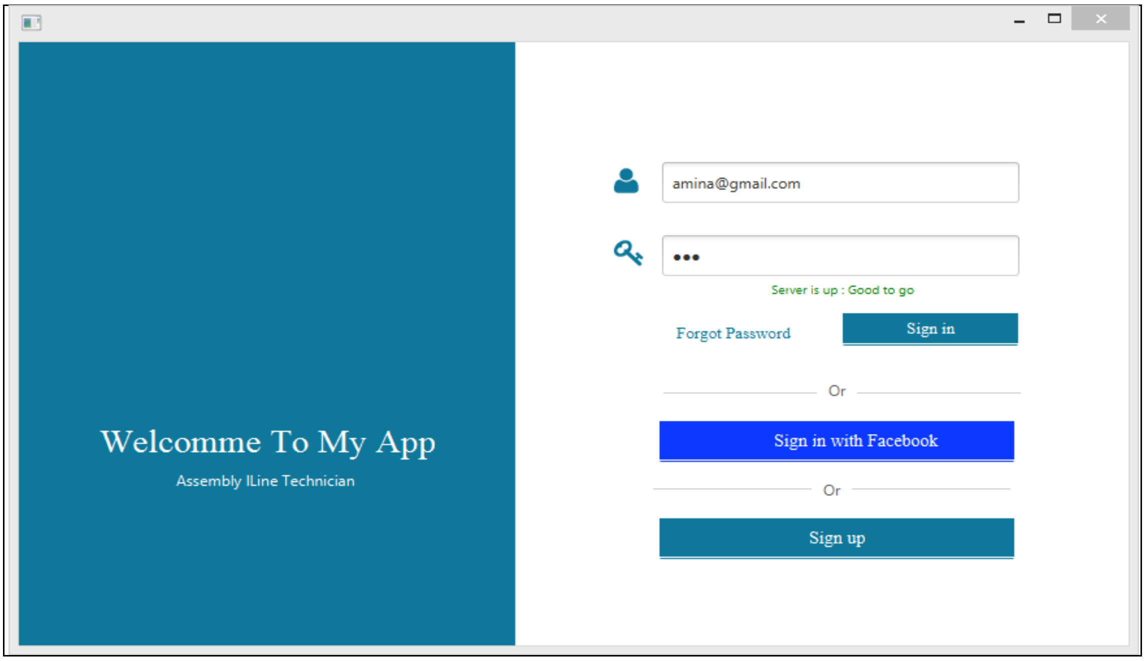


Figure3.14: the authentication technician interface.

4.3-Interface choices technician:

The figure below (Figure3.15) presents Interface choices technician:

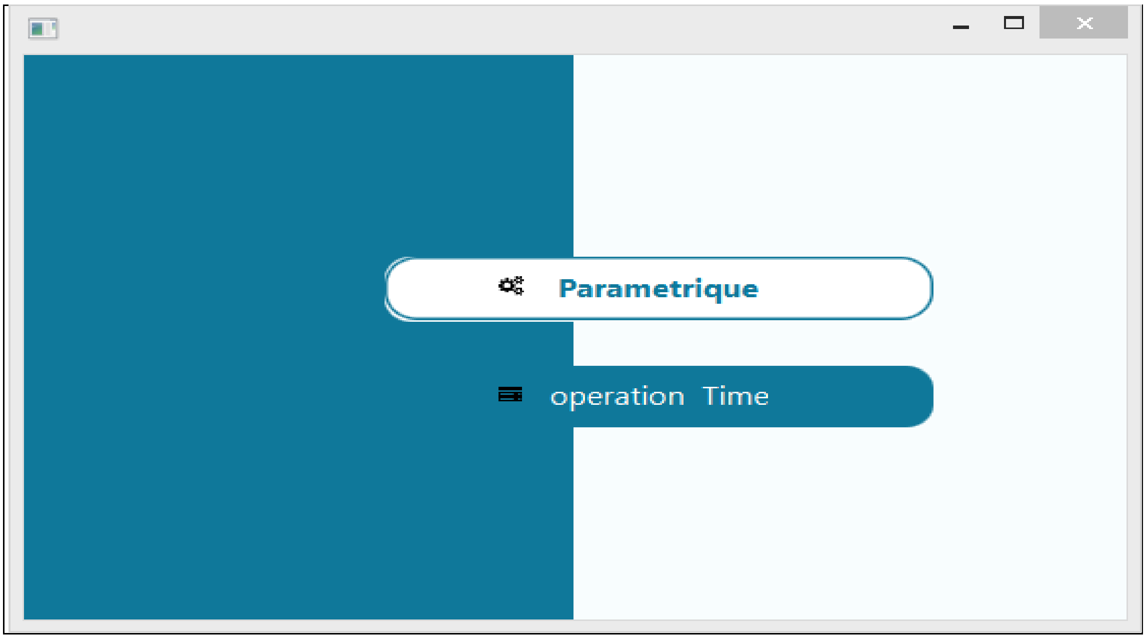
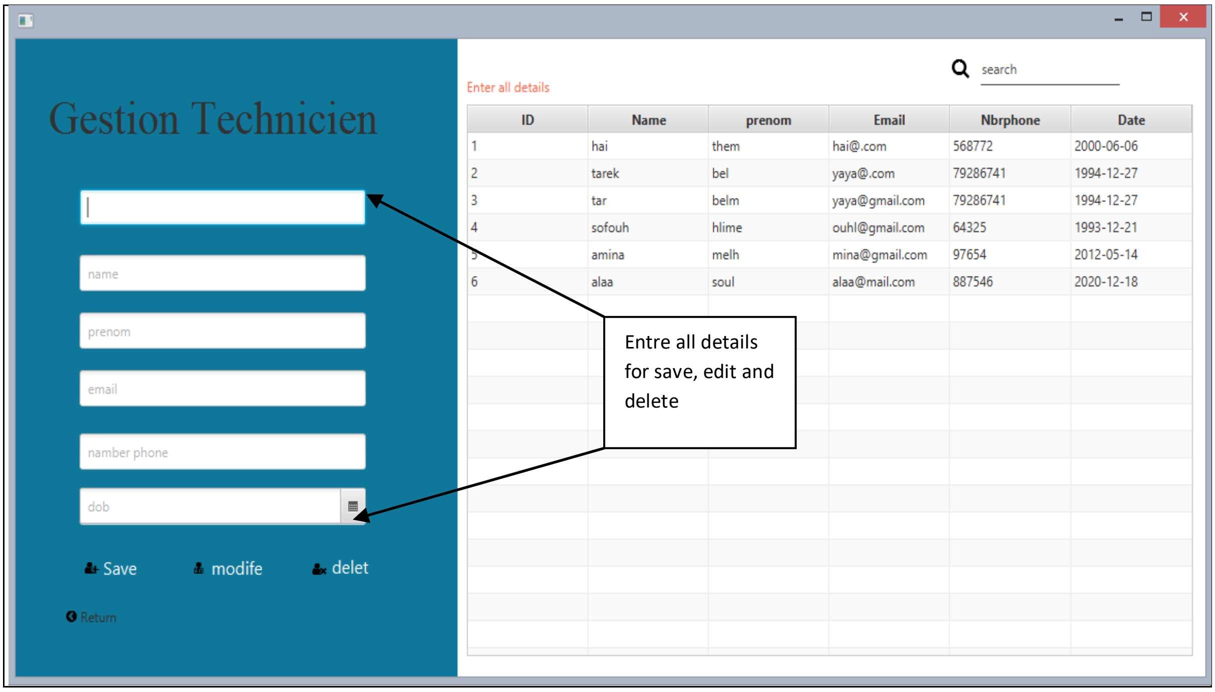


Figure (3.15): the technician choices interface.

4.4- Management technician Interface:

The figure below (Figure3.16) presents Interface management technician:



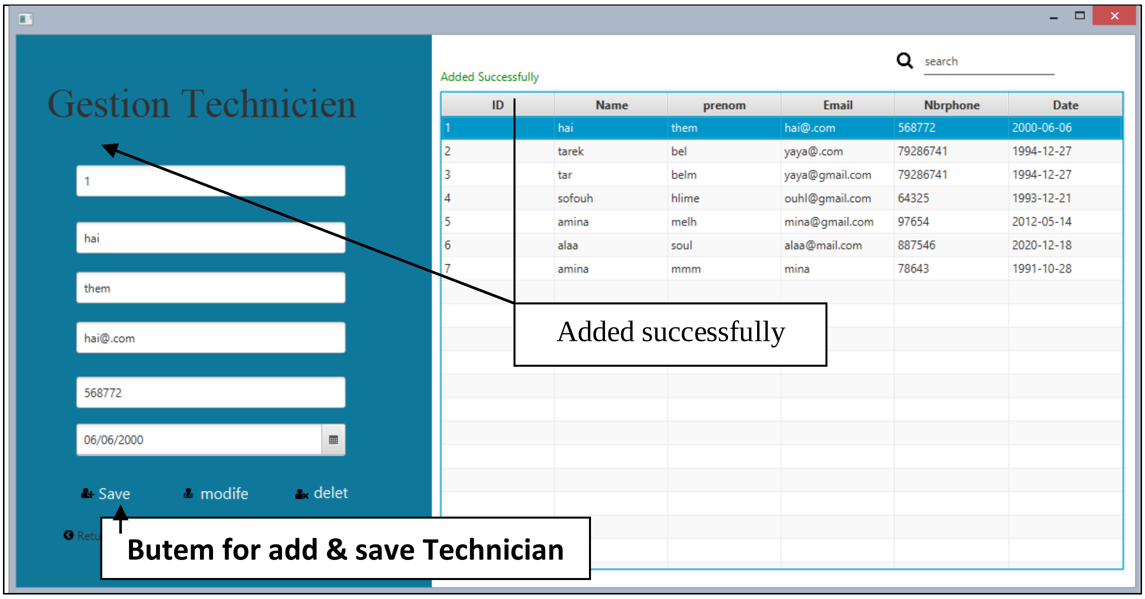
Enter all details

ID	Name	prenom	Email	Nbrphone	Date
1	hai	them	hai@.com	568772	2000-06-06
2	tarek	bel	yaya@.com	79286741	1994-12-27
3	tar	belm	yaya@gmail.com	79286741	1994-12-27
4	sofouh	hlime	ouhl@gmail.com	64325	1993-12-21
5	amina	melh	mina@gmail.com	97654	2012-05-14
6	alaa	soul	alaa@mail.com	887546	2020-12-18

Entre all details for save, edit and delete

Figure (3.16): Interface gestion technician.

The figure below (Figure3.17) presents Interface management add technician:



1
hai
them
hai@.com
568772
06/06/2000

Added Successfully

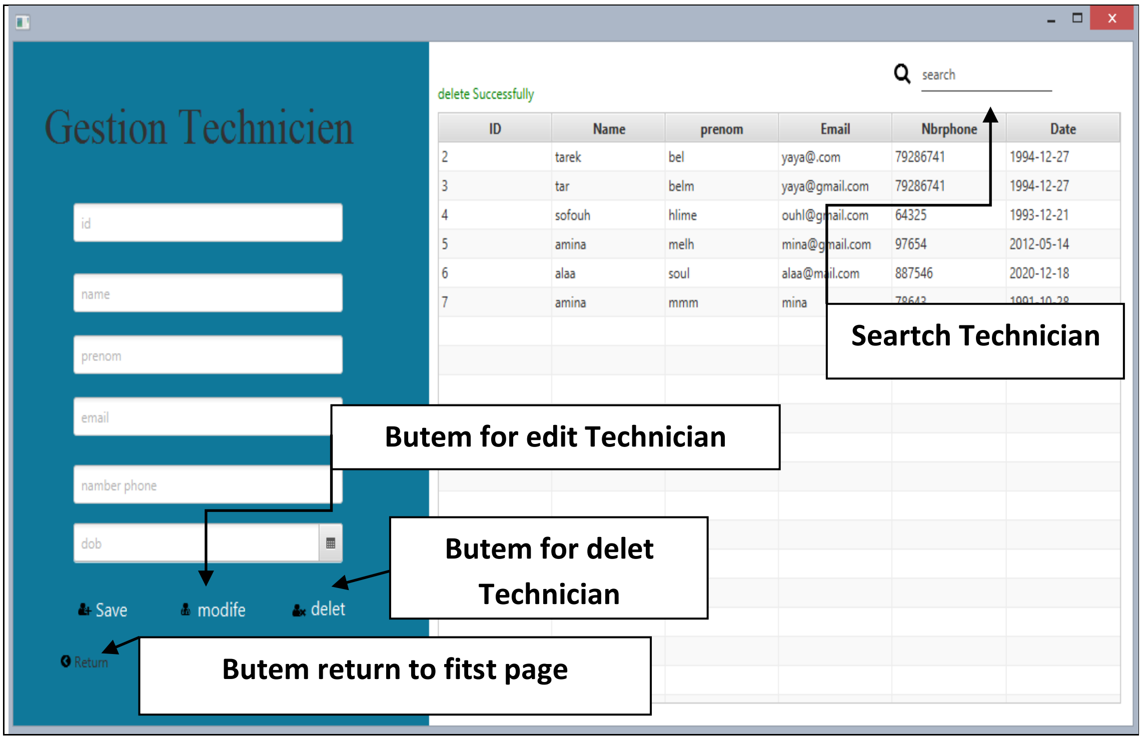
ID	Name	prenom	Email	Nbrphone	Date
1	hai	them	hai@.com	568772	2000-06-06
2	tarek	bel	yaya@.com	79286741	1994-12-27
3	tar	belm	yaya@gmail.com	79286741	1994-12-27
4	sofouh	hlime	ouhl@gmail.com	64325	1993-12-21
5	amina	melh	mina@gmail.com	97654	2012-05-14
6	alaa	soul	alaa@mail.com	887546	2020-12-18
7	amina	mmm	mina	78643	1991-10-28

Added successfully

Butem for add & save Technician

Figure (3.17): Interface management added technician.

The figure below (Figure3.18) presents Interface management delete technician:

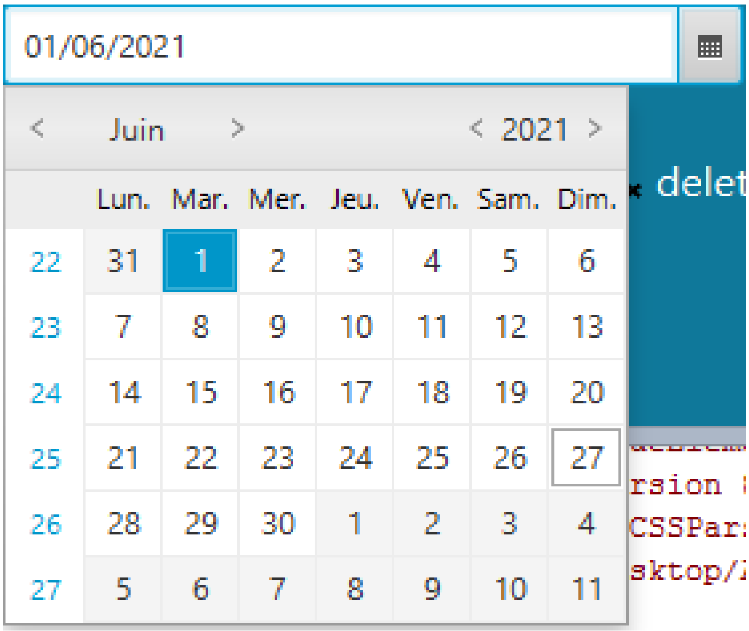


delete Successfully

ID	Name	prenom	Email	Nbrphone	Date
2	tarek	bel	yaya@.com	79286741	1994-12-27
3	tar	belm	yaya@gmail.com	79286741	1994-12-27
4	sofouh	hlime	ouhl@gmail.com	64325	1993-12-21
5	amina	melh	mina@gmail.com	97654	2012-05-14
6	alaa	soul	alaa@mail.com	887546	2020-12-18
7	amina	mmm	mina	78643	1991-10-28

Figure (3.18): Interface gestion delete technician.

The administrator selects the renting period like shown in the following (Figure3.19):



01/06/2021

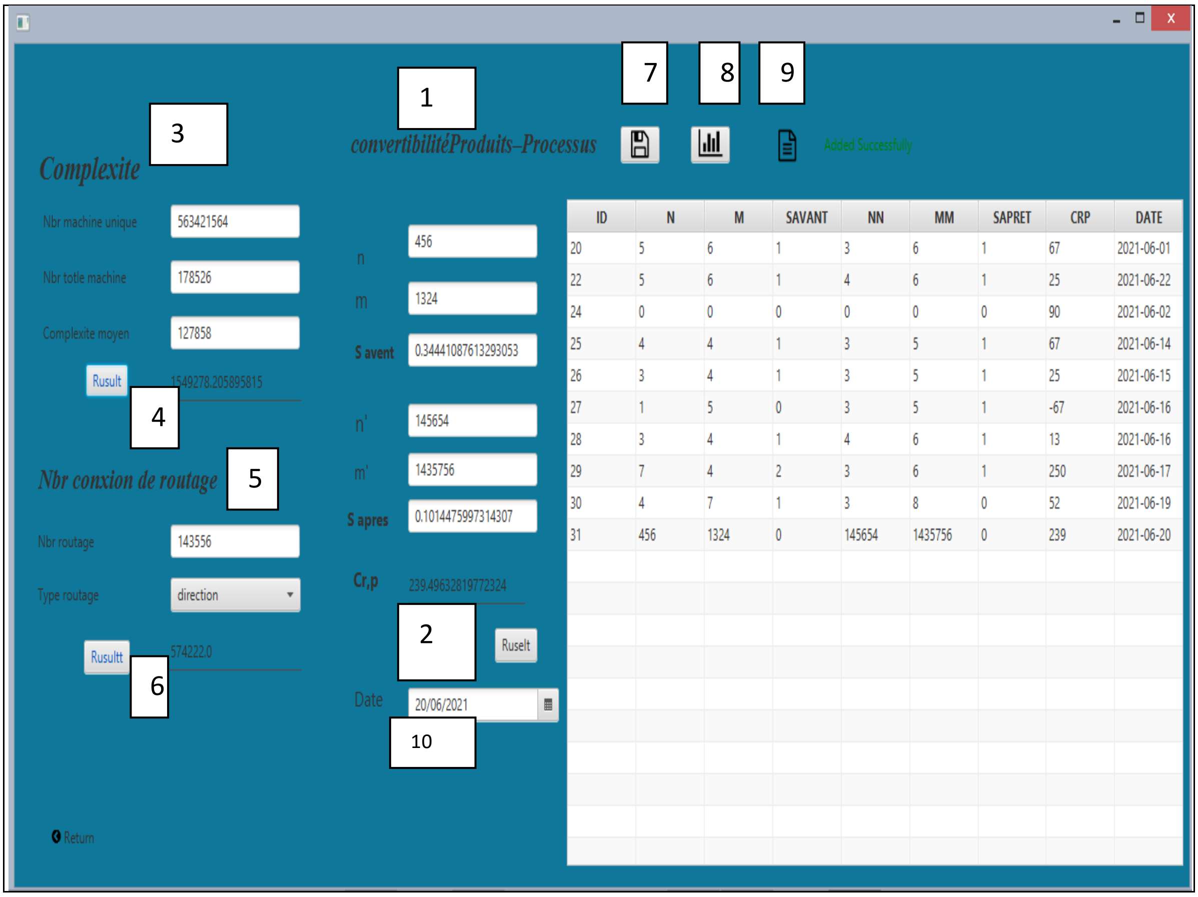
Juin 2021

	Lun.	Mar.	Mer.	Jeu.	Ven.	Sam.	Dim.
22	31	1	2	3	4	5	6
23	7	8	9	10	11	12	13
24	14	15	16	17	18	19	20
25	21	22	23	24	25	26	27
26	28	29	30	1	2	3	4
27	5	6	7	8	9	10	11

Figure (3.19): Date picker.

4.5-parametrique interface:

The figure below (Figure3.20) presents Interface parametrique:



Complexite

Nbr machine unique: 563421564

Nbr totle machine: 178526

Complexite moyen: 127858

1549278.205895815

Nbr conxion de routage

Nbr routage: 143556

Type routage: direction

574222.0

convertibilitéProduits–Processus

n: 456

m: 1324

S avent: 0.34441087613293053

n': 145654

m': 1435756

S apres: 0.1014475997314307

Cr,p: 239.49632819772324

Date: 20/06/2021

Added Successfully

ID	N	M	SAVANT	NN	MM	SAPRET	CRP	DATE
20	5	6	1	3	6	1	67	2021-06-01
22	5	6	1	4	6	1	25	2021-06-22
24	0	0	0	0	0	0	90	2021-06-02
25	4	4	1	3	5	1	67	2021-06-14
26	3	4	1	3	5	1	25	2021-06-15
27	1	5	0	3	5	1	-67	2021-06-16
28	3	4	1	4	6	1	13	2021-06-16
29	7	4	2	3	6	1	250	2021-06-17
30	4	7	1	3	8	0	52	2021-06-19
31	456	1324	0	145654	1435756	0	239	2021-06-20

Figure (3.20): Interface parametrique**.**

Expression function correspond number :

1- Convertibilite produits_processes (Factory production rate) is defined as follows:

- Factory production rate = (Saprès/ Savant) $-1*100$.

- Saprent=the interfaces after the introduction cars.

- Savant= the interfaces before the introduction cars.

2-Result display of factory production rate.

3- Complexity is defined as follows:

-cmplxs = Nu / Nm + (Im * *log*(Nm + 1).

Nu= number machine unique.

Nm= number machine total.

Im=complexity way.

4- Result display of complexity.

5- The number of routing connections is defined as follows

The number of routing connections =$2n + f\ n \sum i$ (i=0, i=n-1).

-f= type routing (unidirectional f=1 and bidirectional f=2).

-n= number routing.

The figure below (Figure3.21) presents choices type of routing:

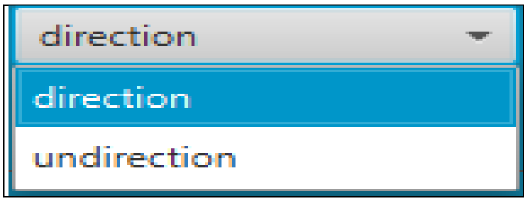


Figure (3.21): choices type of routing.

6- Result display of number of routing connections.

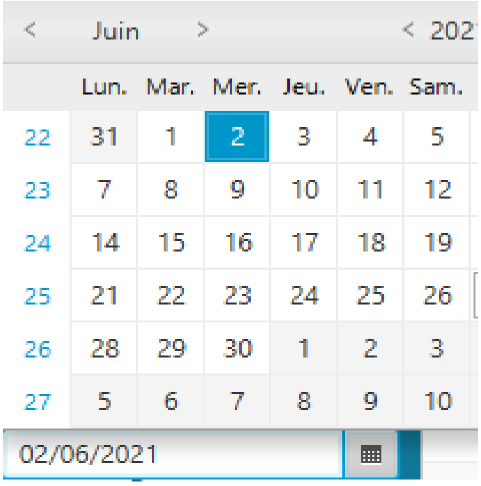
7- Button regestrates the data in the data base.

8-Button views the graph factory production rate line chart.

9-Button views the report factory production rate.

10-Date changes

The technician selects the renting period like shown in the following (Figure3.22):



	Lun.	Mar.	Mer.	Jeu.	Ven.	Sam.
22	31	1	2	3	4	5
23	7	8	9	10	11	12
24	14	15	16	17	18	19
25	21	22	23	24	25	26
26	28	29	30	1	2	3
27	5	6	7	8	9	10

02/06/2021

Figure (3.22): Date picker.

The figure below (Figure3.23) presents Interface line chart of factory production rate:

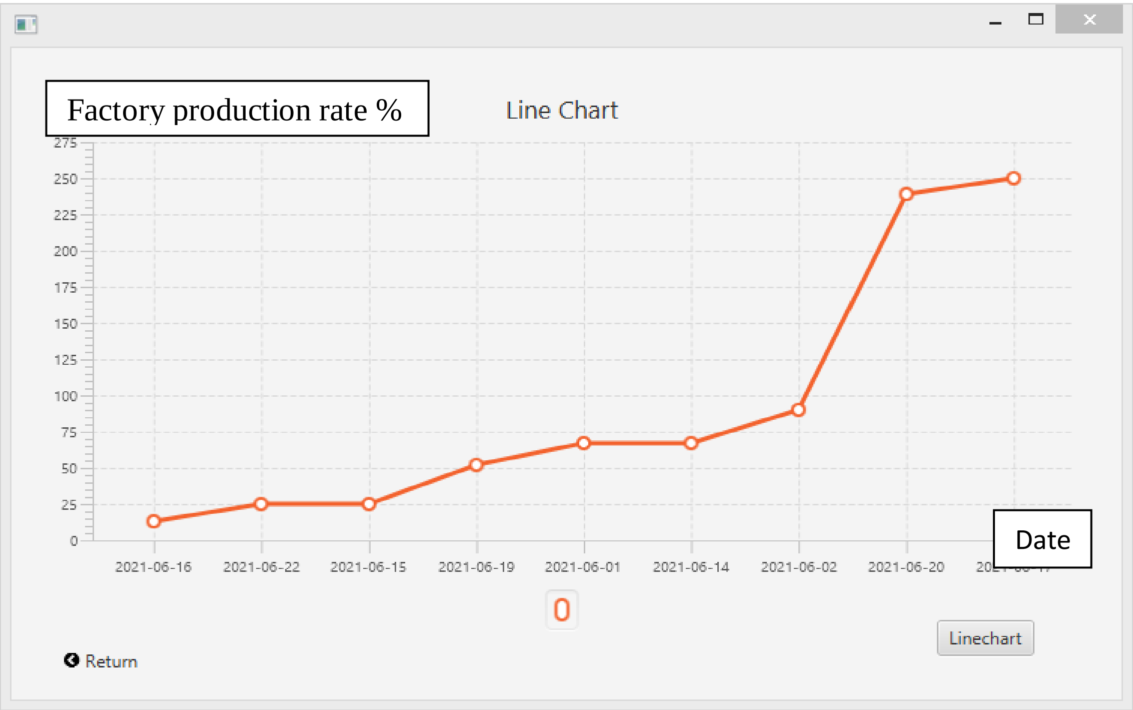
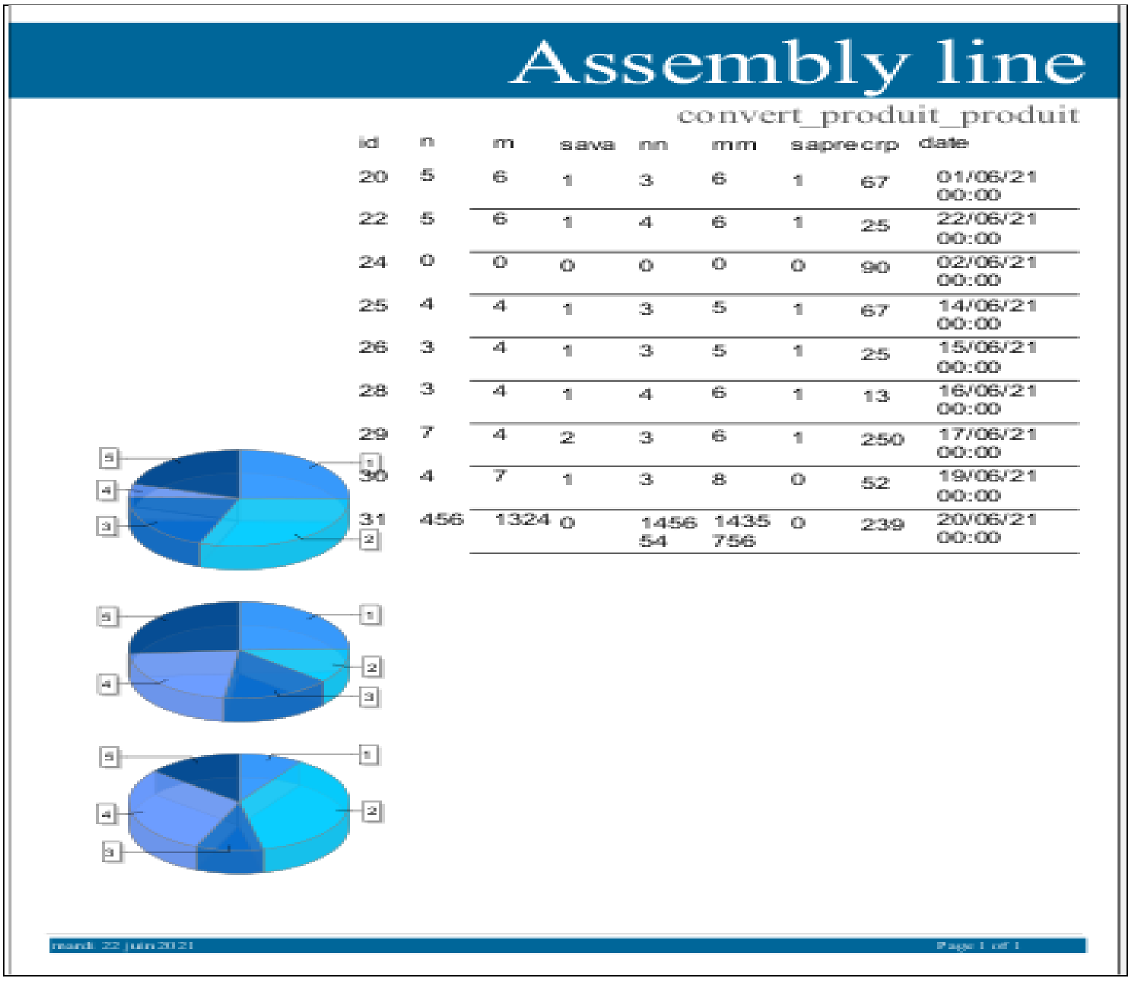


Figure (3.23): Interface line chart of factory production rate.

The figure below (Figure3.24) presents Interface report of factory production rate:



Assembly line

convert_produit_produit

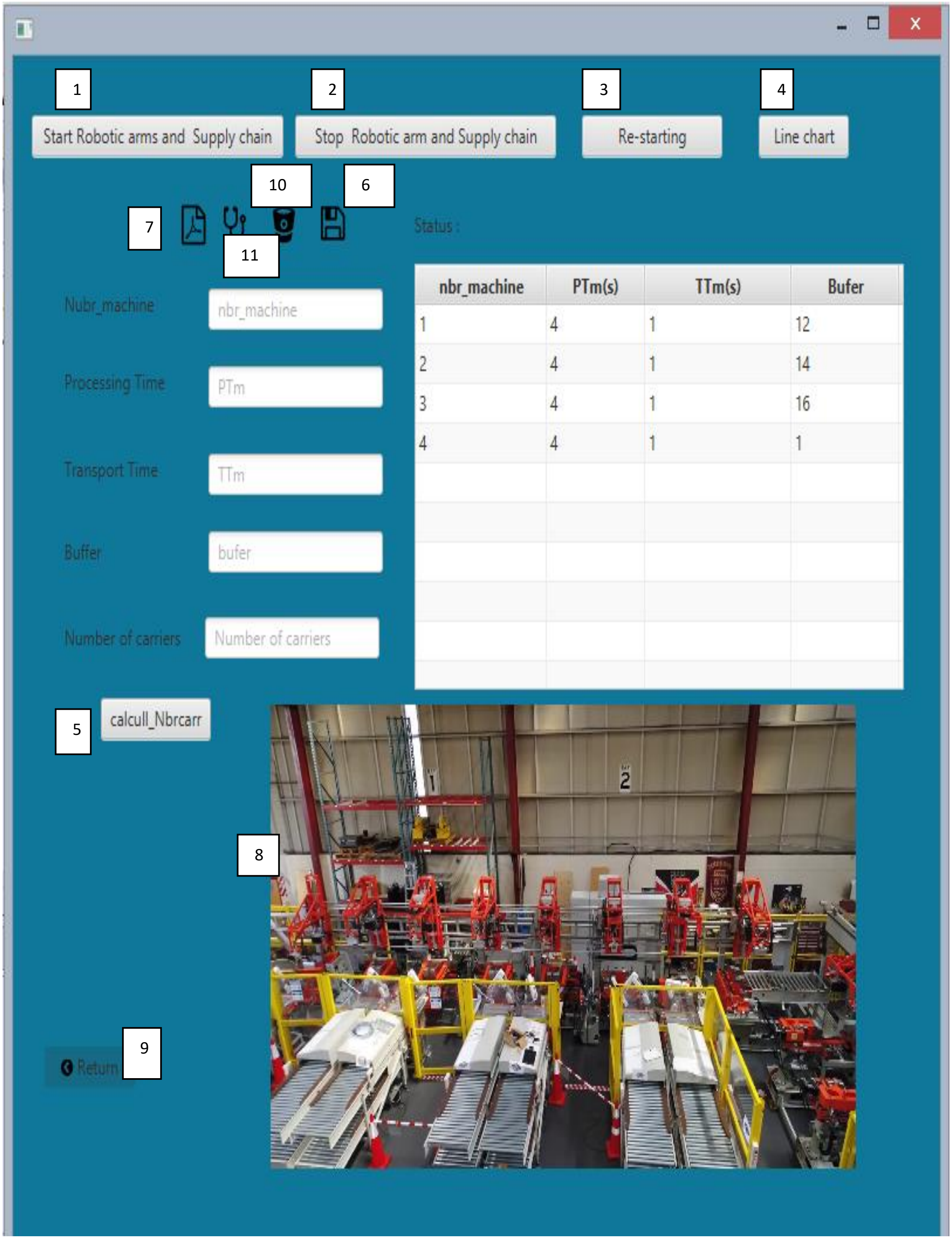
id	n	m	sava	nn	mm	sapre	crp	date
20	5	6	1	3	6	1	67	01/06/21 00:00
22	5	6	1	4	6	1	25	22/06/21 00:00
24	0	0	0	0	0	0	90	02/06/21 00:00
25	4	4	1	3	5	1	67	14/06/21 00:00
26	3	4	1	3	5	1	25	15/06/21 00:00
28	3	4	1	4	6	1	13	16/06/21 00:00
29	7	4	2	3	6	1	250	17/06/21 00:00
30	4	7	1	3	8	0	52	19/06/21 00:00
31	456	1324	0	145654	1435756	0	239	20/06/21 00:00

mardi 22 juin 2021 Page 1 of 1

Figure (3.24): Interface report of factory production rate.

4.6-Cycle Time Function interface:

The figure below (Figure3.25) presents cycle time function:



nbr_machine	PTm(s)	TTm(s)	Bufer
1	4	1	12
2	4	1	14
3	4	1	16
4	4	1	1

Figure (3.25): Interface cycle time.

1- Button start robotic arms and supply chain in same time start video.

2- Button stops robotic arms and supply chain in same time stop video.

3- Button restarts robotic arms and supply chain in same time start video.

4- Button views cycle time line chart.

5- Calcule number carrier.

6- Button register in the database.

7- Button views the graph cycle time line chart

8- Video (media view).

9- Return to the technician choices interface.

10- Button deletes information from database

11- Button edits information from database

The figure below (Figure3.26) presents Interface line chart of cycle time:

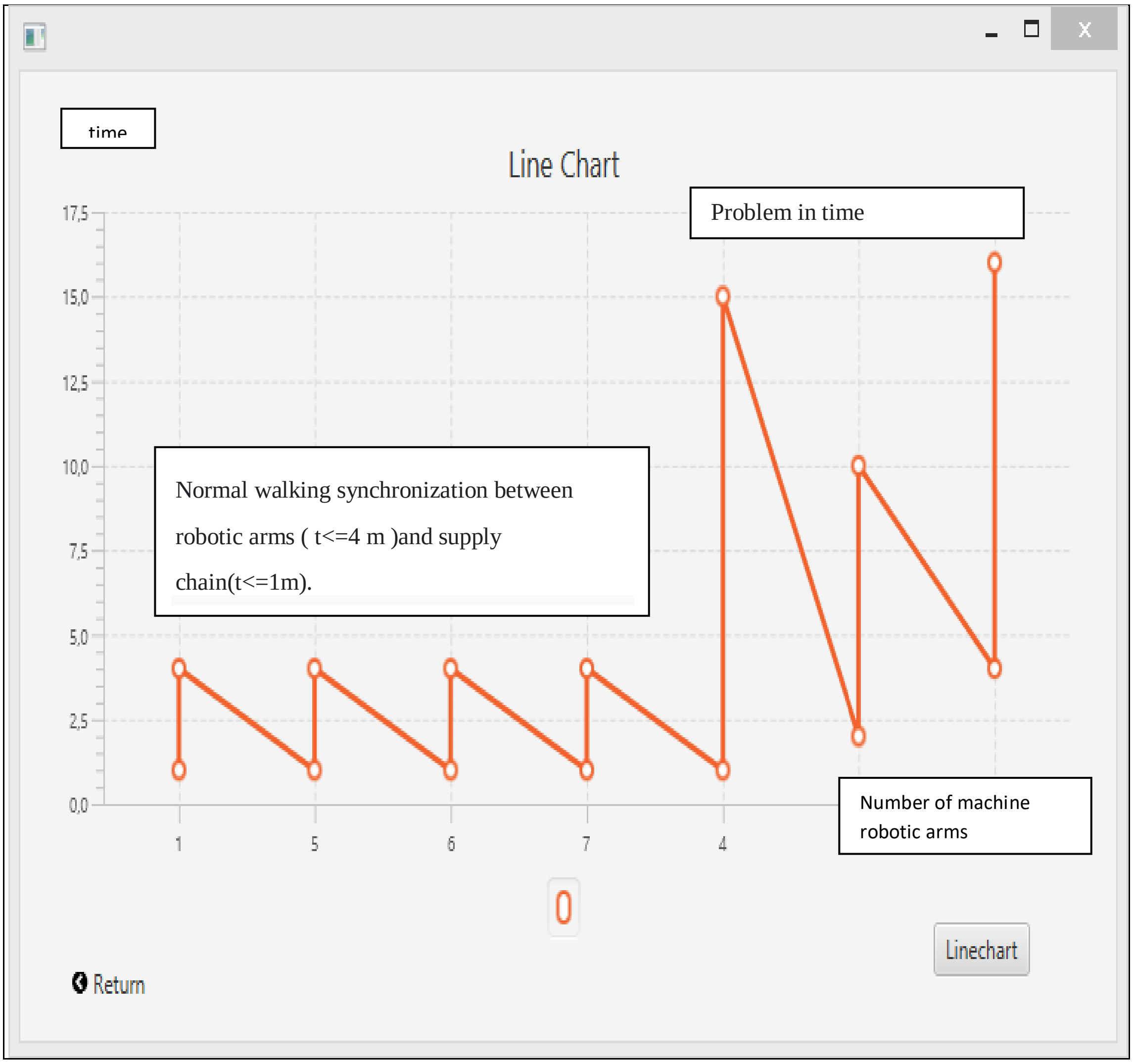
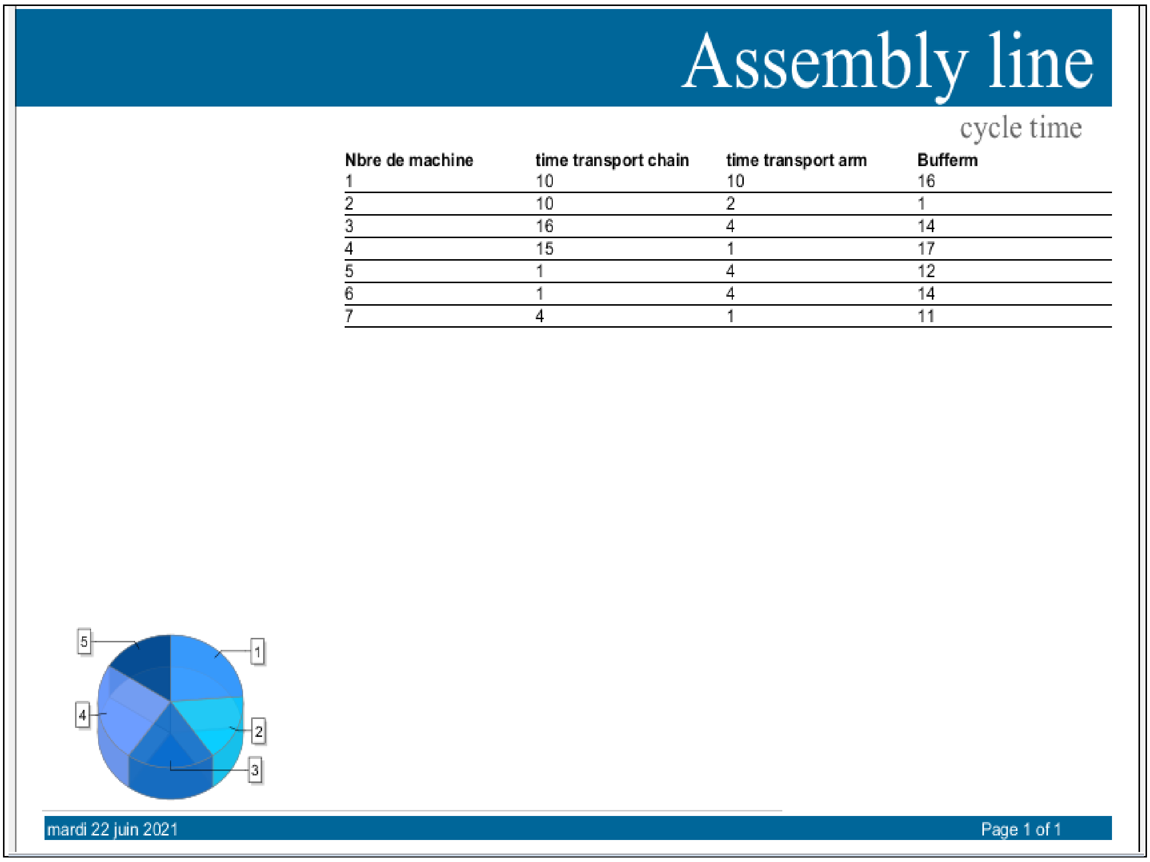

time
Line Chart
Problem in time
17,5
15,0
12,5
10,0
7,5
5,0
2,5
0,0
Normal walking synchronization between robotic arms (t<=4 m)and supply chain(t<=1m).
1
5
6
7
4
Number of machine robotic arms
0
Linechart
Return

Figure (3.26): Interface line chart of cycle time.

The figure below (Figure3.27) presents Interface report of cycle time:



Assembly line

cycle time

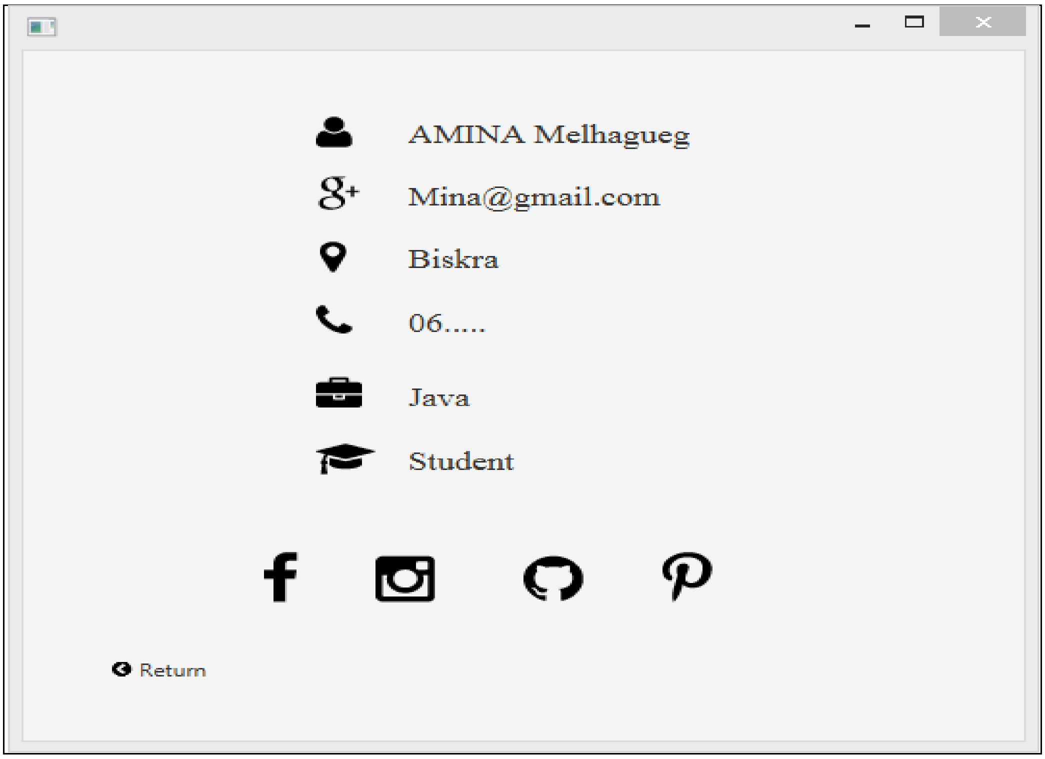
Nbre de machine	time transport chain	time transport arm	Bufferm
1	10	10	16
2	10	2	1
3	16	4	14
4	15	1	17
5	1	4	12
6	1	4	14
7	4	1	11

mardi 22 juin 2021 Page 1 of 1

Figure (3.27): Interface report of cycle time.

4.7- Interface about:

The figure below (Figure3.28) presents Interface about:



AMINA Melhagueg

Mina@gmail.com

Biskra

06.....

Java

Student

Figure (3.28): Interface about.

5-Analysis

Through the use of specification and formal validation of system, we noticed a lot of benefits before implementation, we deduct some benefits:

-Build a bug-free system before build in reality.

-If there are errors, we can find out where the defect is in a short time.

-we select all components and supplies (method, attribute, conditional,…) and define them.

-In specification, we take a long time but this study enables us to build a system in a short time.

-Distribution of services, clarifies tasks, and specifies interfaces.

The following table presents our comparison between study specification and implementation (Table 3.1):

	Specification	**Implementation**
Action	Transition	Method
Automaton time	Automaton	Suitable of function method and condition
CTL	Expretion logique valide and verifie automaton time.	Condition verified structure system building.
System declaration	System name.	Classes name
Time	We take a long time.	We take a short time correspond specification and verification.

Table (3.1): Comparison between specification and implementation.

6-CONCLUSION

In this chapter we have presented the programming languages and the development environments used to build our application to simulate system assembly line cars. Then we illustrated and explained the different features and services of our simulator through GUIs that represent the main interfaces of our simulation application for give precise simulation application by order and organization of information’s, determining of goals.

GENERAL CONCLUSION

GENERAL CONCLUSION

The goal of this work is to simulate and to make formally specify a critical synchronous system as a set of the business process. We apply our contributions in a case related to car installation and manufacturing agency because it provides a set of appropriate internal and external business management processes to have a system that guarantees the best quality and is most suitable for vehicle installation.

Our work follows three phases of life cycle engineering:

First, the design phase uses the semi-formal Models that combine structure description and behavior, such as UML to understand problems, express and model objects independently of any programming language using the following diagrams: diagram of classes that represents the static structure of our system and helps us in the coding phase of our database, use case diagram that provides the functionality of our system from different views (technician/administrator) and then the sequence diagram and the activity diagram where each diagram represented the dynamics of our business process, whether public or private.

Second, the specification phase uses the UPPAAL tool that allows development, verification, validation, and simulation of timed finite-state automaton synchronized, as following:

- Formally description of Assembly line, Robotic arm, supplies chain as the stat machine synchronous automat.
- Declaration of variables and system assembly line by UPPAAL: Business Process Actions. Formal verification of the system modeled properties using the model checking by TCL.
- Validation of Formal system Modeled (simulation).
- Analysis the formal specification gain.

Finally, the implementation phase of our work also exploits different programming languages and assured a coherent matching of the previously made diagrams and specifications to build the most suitable simulation application of a set of public business process management of the assembly line car.

In future work, we plan to improve this project by adding the following features: add other functions, make Ubuntu version and implement this web application for wide use.

BIBLIOGRAPHY

Bibliography

[1] Definition of a business process, website (https://www.appian.com/bpm /definition -of-a-business-process/**),** Visited the10/02/2021.

[2] Definition of a business process, website (https://searchcio.techtarget.com /definition / business -process), Visited the 10/02/2021.

[3] Business process types, website (https://www.feedough.com/what-is-business-process -types-importance-examples/).visited the 12/02/2021.

[4] Characteristics of a business-process, website (cubility.com.au/6-characteristics-of-a-business-process).visited the 15/02/2021.

[5] Characteristics-business-process, website (kaufmanglobal.com/glossary /7-characteristics -business-process).visited the 15/01/2021.

[6] Business Process Model and Notation-BPMN Mark von Rosing, Stephen White, Fred Cummins, Henk de Man.2015.

[7] Facebook definition, website (Facebook.com).visited the 21/02/2021.

[8] Alibaa definition, website (Alibaba.com).visited the 21/02/2021.

[9] Une méthode de conception et de réalisation des processus métiers basés sur les composants et les services Web, KERDOUDI Mohammed Lamine. Master these, 05 /06/2014

[10] Modelisation des processus métiers, website (http://archives.univ-biskra.dz/bitstream /123456789/13223 /1/ben_amar _ali.pdf.), Master these. Visited the 25/02/2012.

[11] Peugeot definition, website (Peugeot.com), visited the 21/02/2021.

[12] Business process modeling, website (https://tallyfy.com/business-processmodeling /), Visited the 28/02/2021.

[13] Business process modeling, website (https://www.cflowapps.com/business-process-modeling/), visited the 28/02/2021

[14] Adaptation et Restructuration de Modèles de Processus workflow dans un Contexte Inter-Organisationnel, BOUKHEDOUMA Saida.2015

[15] Petri net, website (https://www.techopedia.com/definition/18765/petri-net), visited the 25/02/2021.

[16] Business processes management technology, website (https://www. sutherlandglobal.com/our-thinking/what-is-bpm-technology),visited the 24/02/2021.

[17] Business management consulting, website (https://www.lmtecdigitalsolutions. com /en/ business-management-consulting), visited the 26/02/2021.

[18] Définition Réseau_de_Petri, website **(**http://www.morere.eu/IMG/pdf/cours_ petri2.pdf).années 2002.

[19] UML diagram types examples (https://creately.com/blog/diagrams/uml-diagram-types-examples/), visited the 13/05/2021.

[20] Integrated Definition Method 3 (IDEF3), website (https://www.idef.com), visited the 28/02/2021.

[21] Definition Π-calculus, website (https://www.brics.dk/LS/98/4/BRICS-LS-98-4.pdf), visited the 03/03/2021.

[22] Business Process Definition Metamodel, website (https://www.Omg.org /spec/BPDM/1.0/ About-BPDM/), visited the 03/03/2021.

[23] La norme du business process management, website (https://www.bpms .info/bpmn-la-norme-du-bpm/), visited the24/02/2021.

[24] Definition XPDL, website (https://www.omg.org /bpmn/Documents/XPDL _BPMN.pdf), Visited the 26/02/2021.

[25] Business process management, website (https://www.sweetprocess.com/business-process-management/), visited the 07/03/2021.

[26] Business process lifecycle, website (https://www.heflo.com/blog/bpm/business-process-lifecycle/), visited the 14/04/2021.

[27] Business process management overview, website (https://kissflow .com/workflow/bpm/ business-process-management-overview/), Visited the 15/03/2021.

[28] Figure of Timed Automaton, website (https://www.researchgate.net/figure /Exemple-dautomate-temporise-Cet-automate-comporte-trois-localites-start-qui-est-la_fig6_281984846), Visited the 17/04/2021.

[29] Definition business process, website (https://www.aiim.org/what-is-bpm), visited the 27/04/2021.

[30] Figure AEF, website (https://tel.archives-ouvertes.fr/tel-00720666/document), visited the 17/04/2021.

[31] Figure example of Business Process non-exactable, website (http://blog.maxconsilium.com /2013/09/bpmn-20-models-part-1.html), Visited the 17/04/2021.

[32] Un Framework pour la gestion et l'interaction des processus métiers intra et inter entreprises, OUAAR. Hanane, PhD these, 2010.

[33] Iphone definition, website (futura-sciences.com/tech/definitions/apple-iphone-5265), visited the 21/02/2021.

[34] Definition information system, website (https://www.britannica.com /topic/information-system), visited the 28/06/2021.

[35] Business process, website (https://kissflow.com/workflow/bpm/business-process/), visited the 27/04/2021.

[36] Business process design, website (https://checkify.com/blog/business-process-design/), visited the 24/02/2021.

[37] What is a workflow, website (https://kissflow.com/workflow/what-is-a-workflow/), visited the 27/04/2021.

[38] Service web definition, website (https://www.lebigdata.fr/services-web-definition), visited the 17/04/2021.

[39] Cours architectures orientées service, kerdoudi md Lamine 2020/2021.

[40] Formal methods model, website (https://ecomputernotes.com/software-engineering/formal-methods-model.), Visited the 17/04/2021.

[42] Business Process Verification: The Application of Model Checking and Time Automata; Luis E. Mendoza Morales. 2014.

[43] Definition Timed Automaton, website (http://www.scielo.edu.uy/scielo.php), Visited the 18/04/2021.

[44] Temporal logic definition, website (https://plato.stanford.edu/entries/logic-temporal/), visited the 18/04/2021.

[45] Definition finished-state automaton, website (https://elearning.univ bejaia.dz / pluginfile. php), Visited the 18/04/2021.

[46] Test and diagnosis of discrete event systems using Petri nets, Marco POCCI.Visited the 13/06/2021.

[47] Definition of modelio, website (hmodelio.org/downloads/download-modelio.html),Visited the 15/05/2021.

[48] Definition of modelio, website (https://www.cpswarm.eu/index.php/frevo-and-modelio/) , Visited the 15/05/2021.

[49] Definition of jaspersoft studio, website (https://community.jaspersoft.com /wiki/introduction-jaspersoft-studio),Visited the 16/06/2021.

[50] Robotic arms guide, website (https://uk.rs-online.com/web/generalDisplay .html?id=ideas-and-advice/robotic-arms-guide), Visited the 24/04/2021.

[51] Automated assembly line robotic arms, website (iptmajor.weebly.com/automated-assembly -line-robotic-arms), Visited the 25/04/2021.

[52] Definition assembly line cars, website (https://media.cheggcdn.com), visited the 15/04/2021.

[53] Definition of supply chain, website (https://www.tandfonline.com/doi/abs /10.1080 /00207543.2011.593052?j), Visited the 25/04/2021.

[54] Comment les voitures sont elles fabriquees, website (https://miec.fr/comment-voitures-sont-elles-fabriquees/), Visited the 02/06/2021.

[55] Les differentes etapes de construction dune voiture, website (https://projector-global. net/quotidien/les-differentes-etapes-de-construction-dune-voiture/), Visited the 02/06/2021.

[56] Meriem Lafou (Contribution à la conception de systèmes, d'assemblage automobile, performants, Pérennesetinnovants pardes) ,2016.

[57] Machine assembly, website (https://www.chegg.com/homework-help/questions-and-answers/figure-shows-puma-programmable-universal-machine-assembly-programmable-universal-manipulat-q32041760), visited the 03/06/2021.

[58] Benefits of business process-management, website (https://insights.diligent .com/business-structures/six-benefits-of-business-process-management/), Visited the 18/02/2021.

[59] JavaFX definition, website (https://www.theserverside.com/definition/JavaFX), visited the 15/06/2021.

[60] Scene builder definition, website (https://riptutorial.com/javafx/topic/5445/scene-builder), visited the 15/06/2021.

[61] Definition Cascading Style Sheets, website (https://www.futura-sciences.com/tech /definitions/internet-css-4050/), Visited the 17/06/2021.

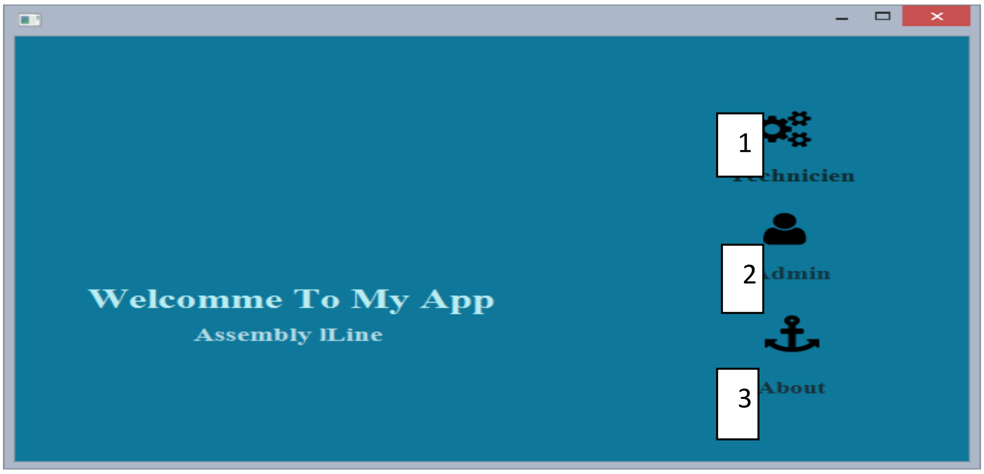
[62] xampp definition, website (http://www.standard-du-web.com/xampp.php.), Visited the 17/06/2021.

Annex

Annex

1. The access interface:

The figure below Figure 1 presents the main access interface:



Welcomme To My App

Assembly ILine

Figure1: *The main interface.*

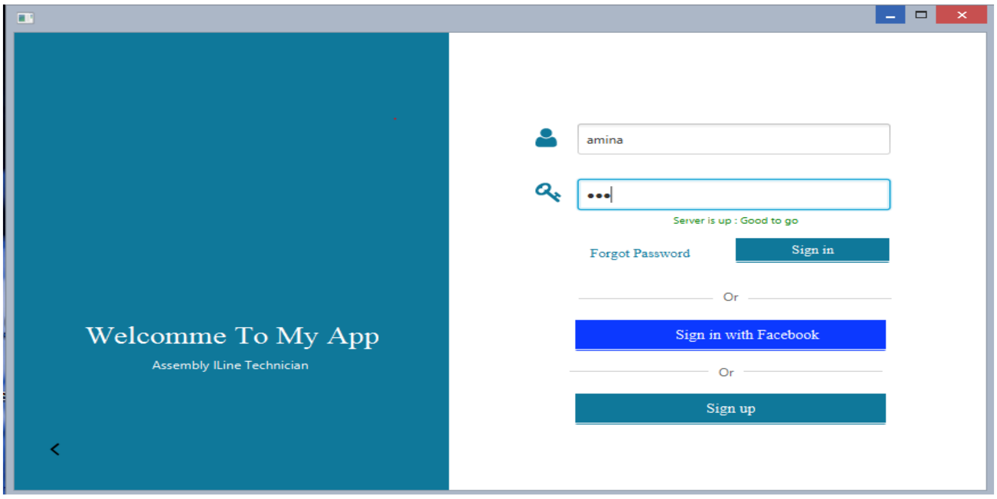
1- Click to go to the authentication technician interface.

2- Click to go to the authentication admin interface

3- Click to go to the about interface.

2. The authentication technician interface

The figure below (Figure2) presents the authentication technician interface



Welcomme To My App

Assembly ILine Technician

amina

•••

Server is up : Good to go

Figure3.14: *The authentication technician interface.*

The figure below (Figure3) presents Interface choices technician:

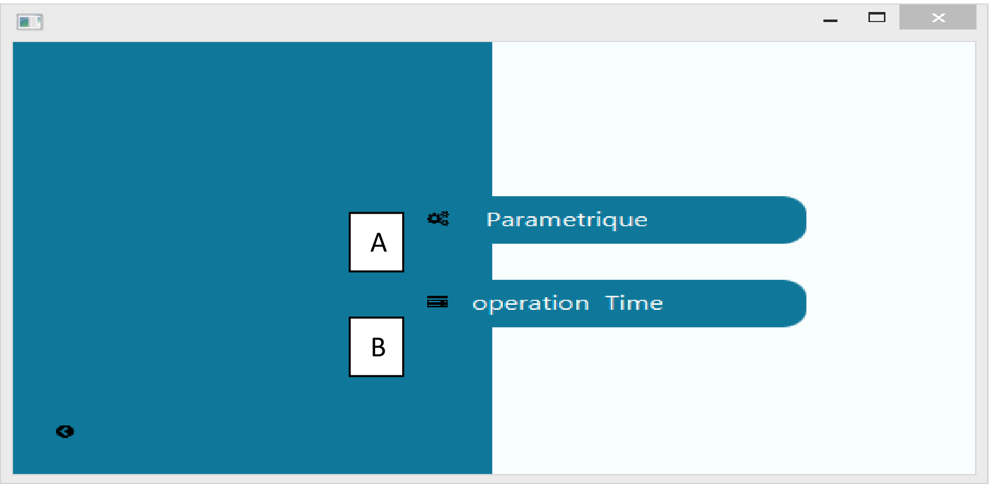
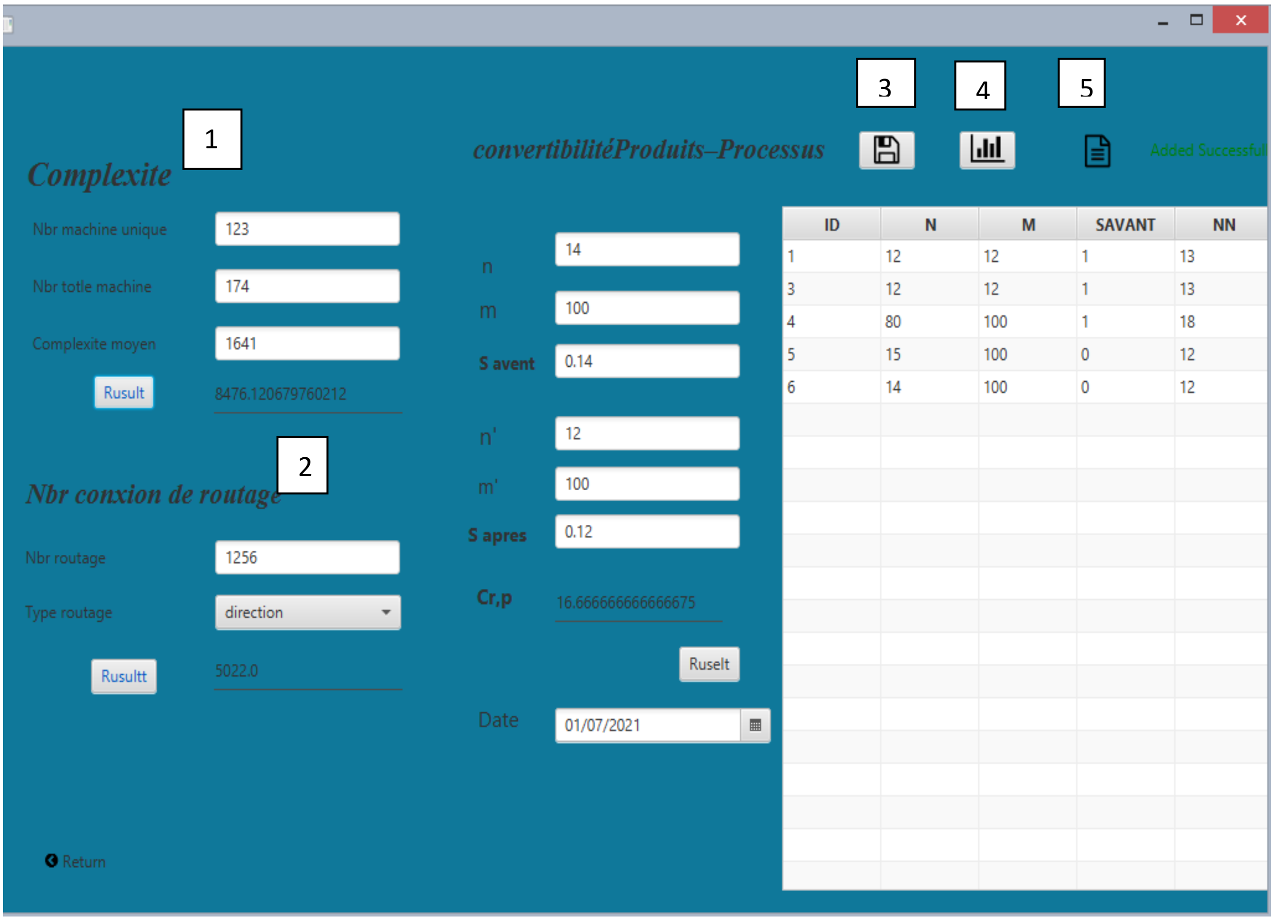


Figure (3): *The technician choices interface.*

A-Setting interface:

The figure below (Figure 4) presents the setting Interface “parametrique”:



Complexite

Nbr machine unique 123

Nbr totle machine 174

Complexite moyen 1641

8476.120679760212

Nbr conxion de routage

Nbr routage 1256

Type routage direction

5022.0

convertibilitéProduits–Processus

n 14

m 100

S avent 0.14

n' 12

m' 100

S apres 0.12

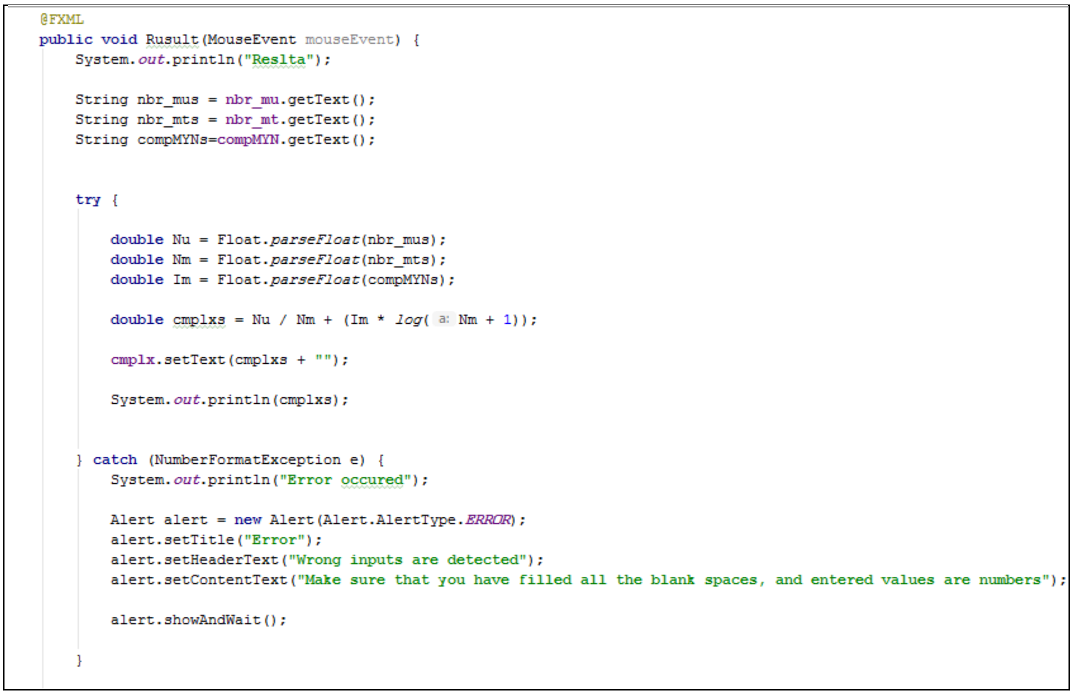
Cr,p 16.666666666666675

Date 01/07/2021

ID	N	M	SAVANT	NN
1	12	12	1	13
3	12	12	1	13
4	80	100	1	18
5	15	100	0	12
6	14	100	0	12

Figure (4): *Setting Interface.*

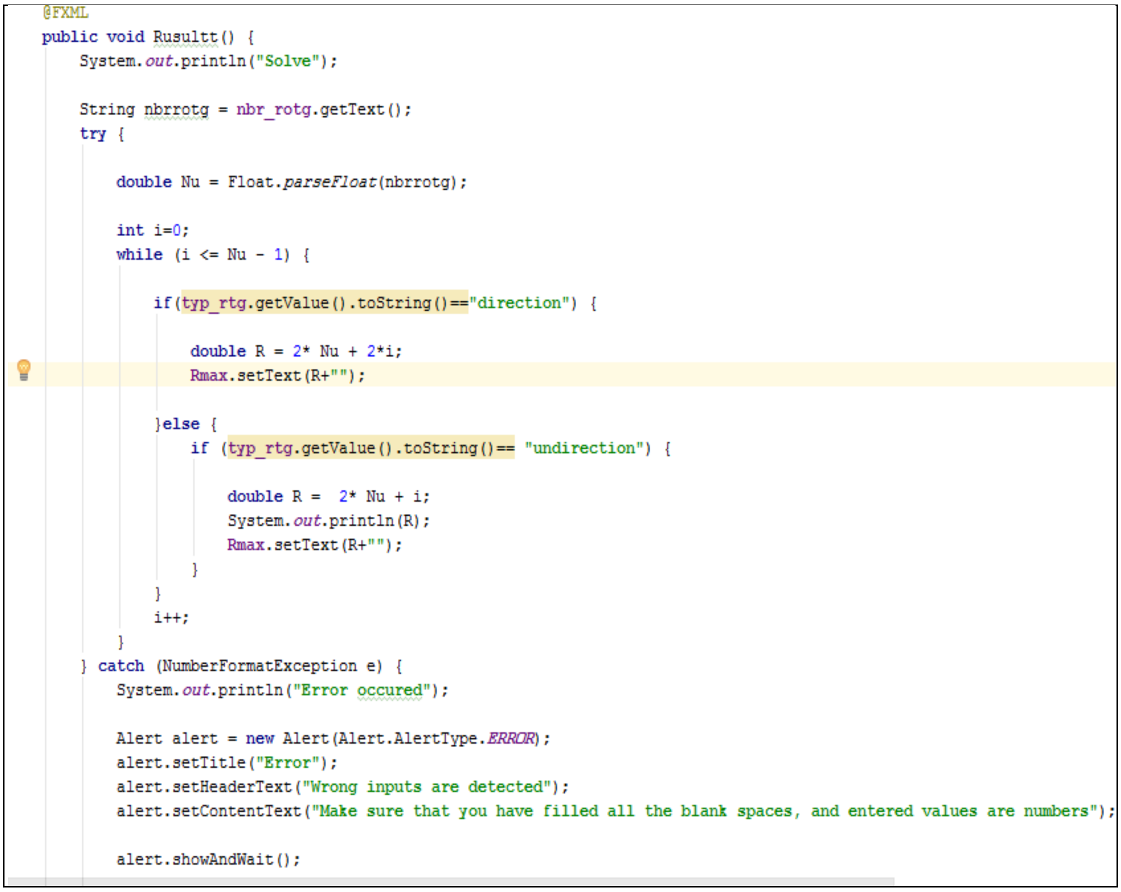
The figure below shows the source code that calculus complexity:



```
@FXML
public void Rusult(MouseEvent mouseEvent) {
    System.out.println("Reslta");

    String nbr_mus = nbr_mu.getText();
    String nbr_mts = nbr_mt.getText();
    String compMYNs=compMYN.getText();


    try {

        double Nu = Float.parseFloat(nbr_mus);
        double Nm = Float.parseFloat(nbr_mts);
        double Im = Float.parseFloat(compMYNs);

        double cmplxs = Nu / Nm + (Im * log( a: Nm + 1));

        cmplx.setText(cmplxs + "");

        System.out.println(cmplxs);


    } catch (NumberFormatException e) {
        System.out.println("Error occured");

        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error");
        alert.setHeaderText("Wrong inputs are detected");
        alert.setContentText("Make sure that you have filled all the blank spaces, and entered values are numbers");

        alert.showAndWait();

    }
```

Figure (5): *The source code calculus complexity.*

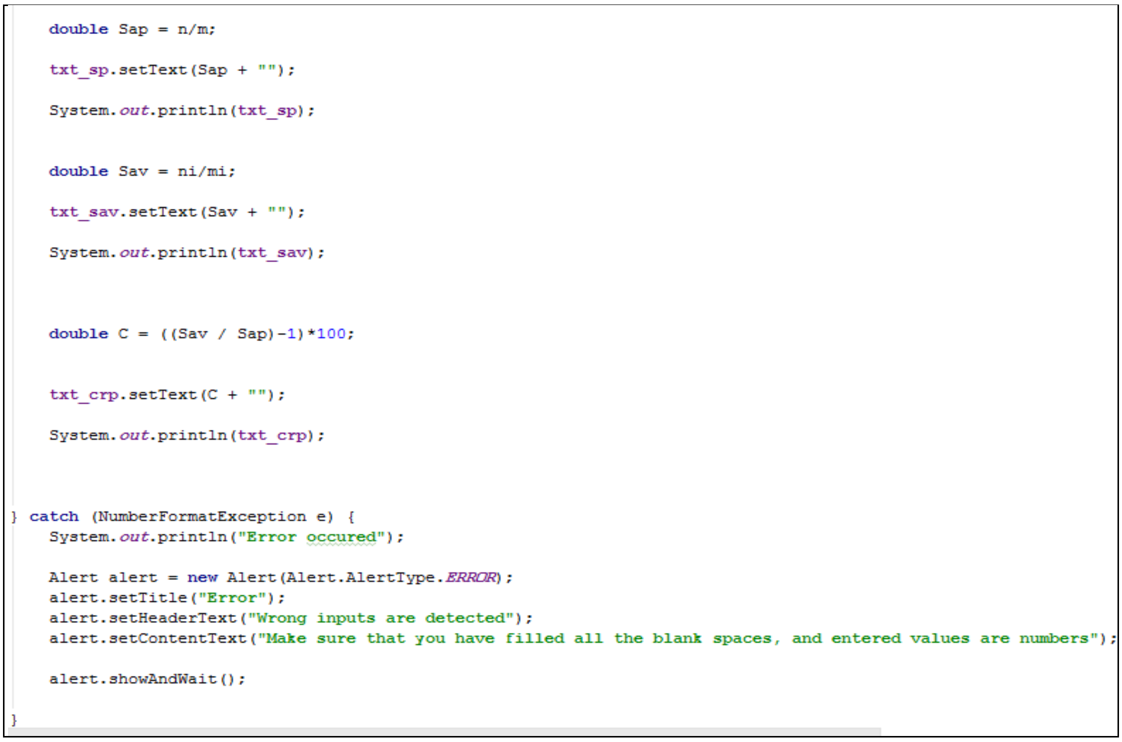
The figure below shows the source code that calculus number of routine connation:



```
@FXML
public void Rusultt() {
    System.out.println("Solve");

    String nbrrotg = nbr_rotg.getText();
    try {

        double Nu = Float.parseFloat(nbrrotg);

        int i=0;
        while (i <= Nu - 1) {

            if(typ_rtg.getValue().toString()=="direction") {

                double R = 2* Nu + 2*i;
                Rmax.setText(R+"");

            }else {
                if (typ_rtg.getValue().toString()== "undirection") {

                    double R =  2* Nu + i;
                    System.out.println(R);
                    Rmax.setText(R+"");
                }
            }
            i++;
        }
    } catch (NumberFormatException e) {
        System.out.println("Error occured");

        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error");
        alert.setHeaderText("Wrong inputs are detected");
        alert.setContentText("Make sure that you have filled all the blank spaces, and entered values are numbers");

        alert.showAndWait();
```

Figure (6): *The source code to calculate the routine's number of connation.*

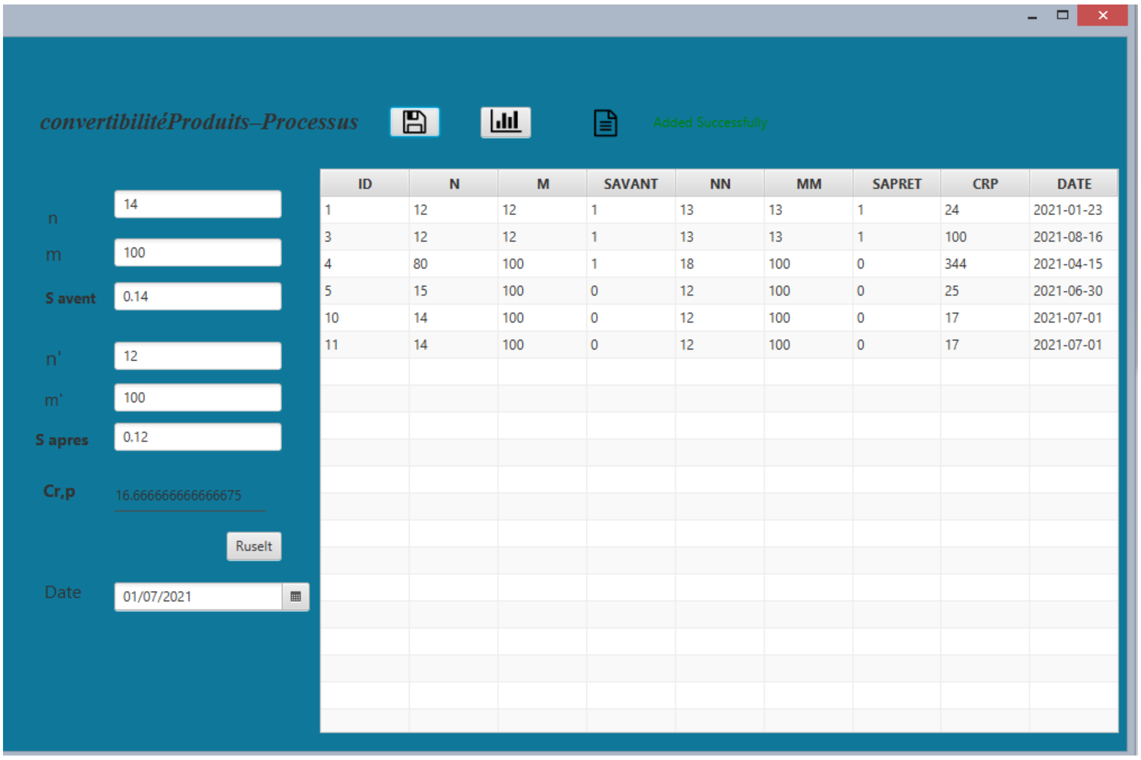
The figure bellow shows the source code to calculate the conv produits_processes:



```
    double Sap = n/m;

    txt_sp.setText(Sap + "");

    System.out.println(txt_sp);


    double Sav = ni/mi;

    txt_sav.setText(Sav + "");

    System.out.println(txt_sav);



    double C = ((Sav / Sap)-1)*100;


    txt_crp.setText(C + "");

    System.out.println(txt_crp);



} catch (NumberFormatException e) {
    System.out.println("Error occured");

    Alert alert = new Alert(Alert.AlertType.ERROR);
    alert.setTitle("Error");
    alert.setHeaderText("Wrong inputs are detected");
    alert.setContentText("Make sure that you have filled all the blank spaces, and entered values are numbers");

    alert.showAndWait();

}
```

Figure (7): *The source code to calculate the conv produits_processes.*

The figure below (Figure 8) presents Interface that add calculates the conv produits_processes.



n: 14

m: 100

S avent: 0.14

n': 12

m': 100

S apres: 0.12

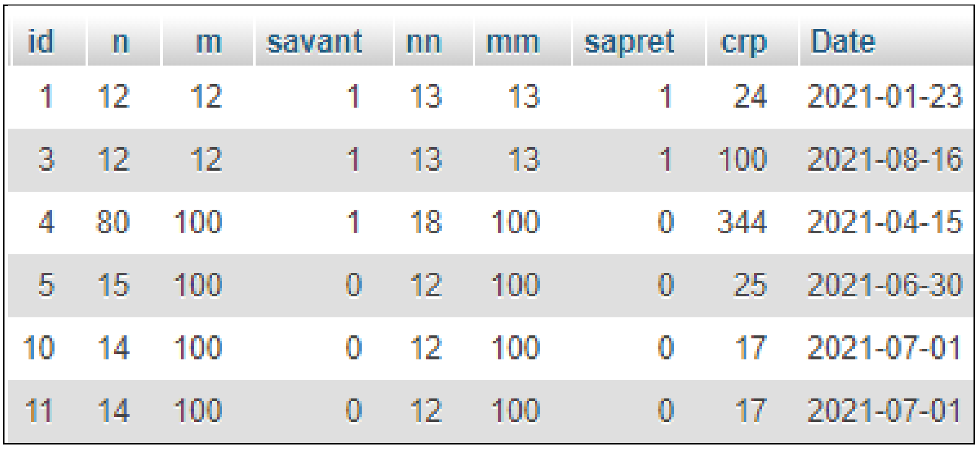
Cr,p: 16.666666666666675

Date: 01/07/2021

ID	N	M	SAVANT	NN	MM	SAPRET	CRP	DATE
1	12	12	1	13	13	1	24	2021-01-23
3	12	12	1	13	13	1	100	2021-08-16
4	80	100	1	18	100	0	344	2021-04-15
5	15	100	0	12	100	0	25	2021-06-30
10	14	100	0	12	100	0	17	2021-07-01
11	14	100	0	12	100	0	17	2021-07-01

Figure (8): *Interface add calculates the conv produits_processes.*

The figure below (Figure 9) presents database of calculates the conv produits_processes.



id	n	m	savant	nn	mm	sapret	crp	Date
1	12	12	1	13	13	1	24	2021-01-23
3	12	12	1	13	13	1	100	2021-08-16
4	80	100	1	18	100	0	344	2021-04-15
5	15	100	0	12	100	0	25	2021-06-30
10	14	100	0	12	100	0	17	2021-07-01
11	14	100	0	12	100	0	17	2021-07-01

Figure (9): *Add in the database the conv produits_processes.*

The figure below (Figure 10) presents Interface line chart of setting:

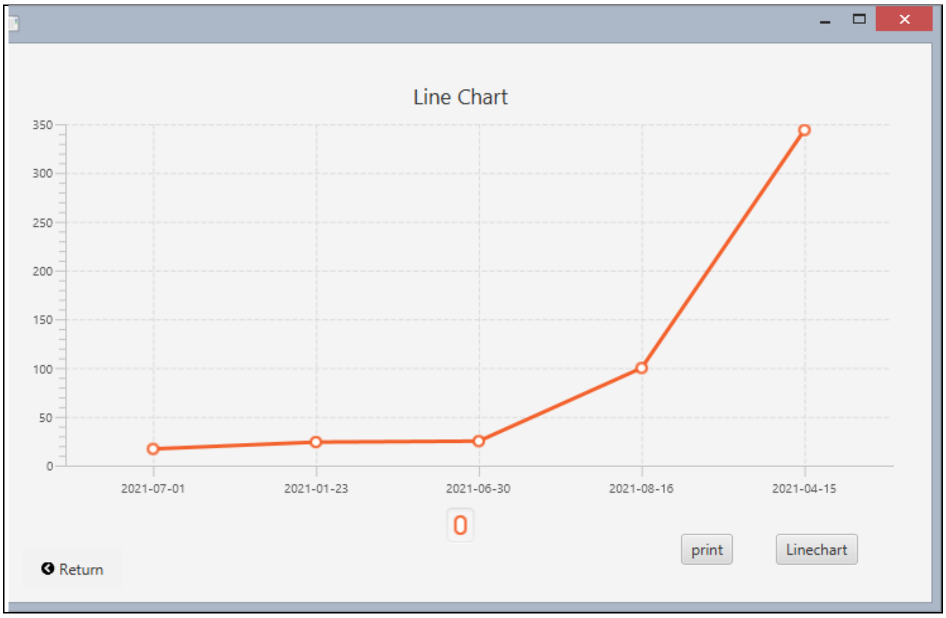
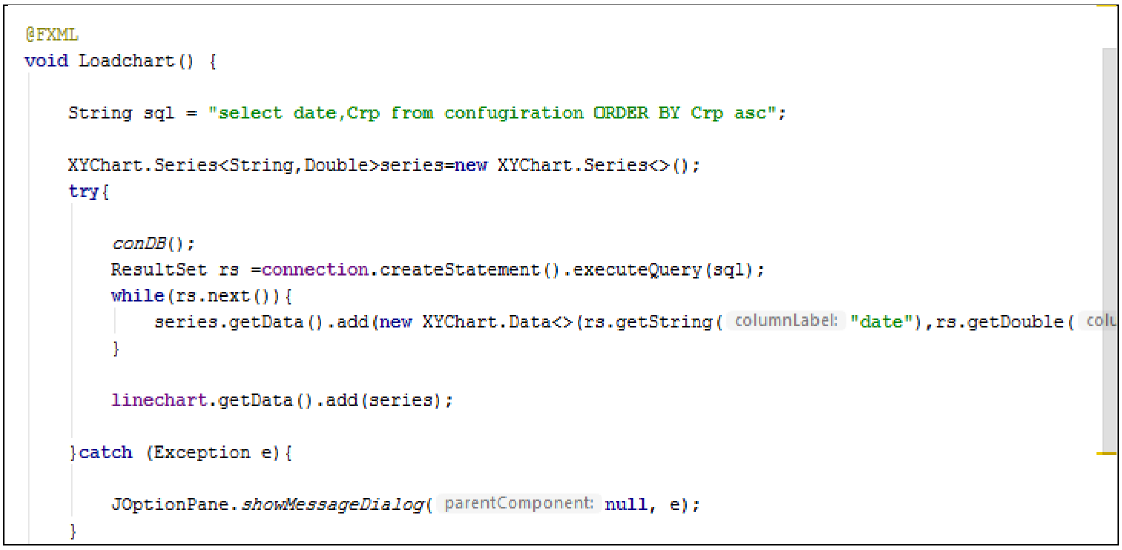


Figure (10): *Interface line chart.*

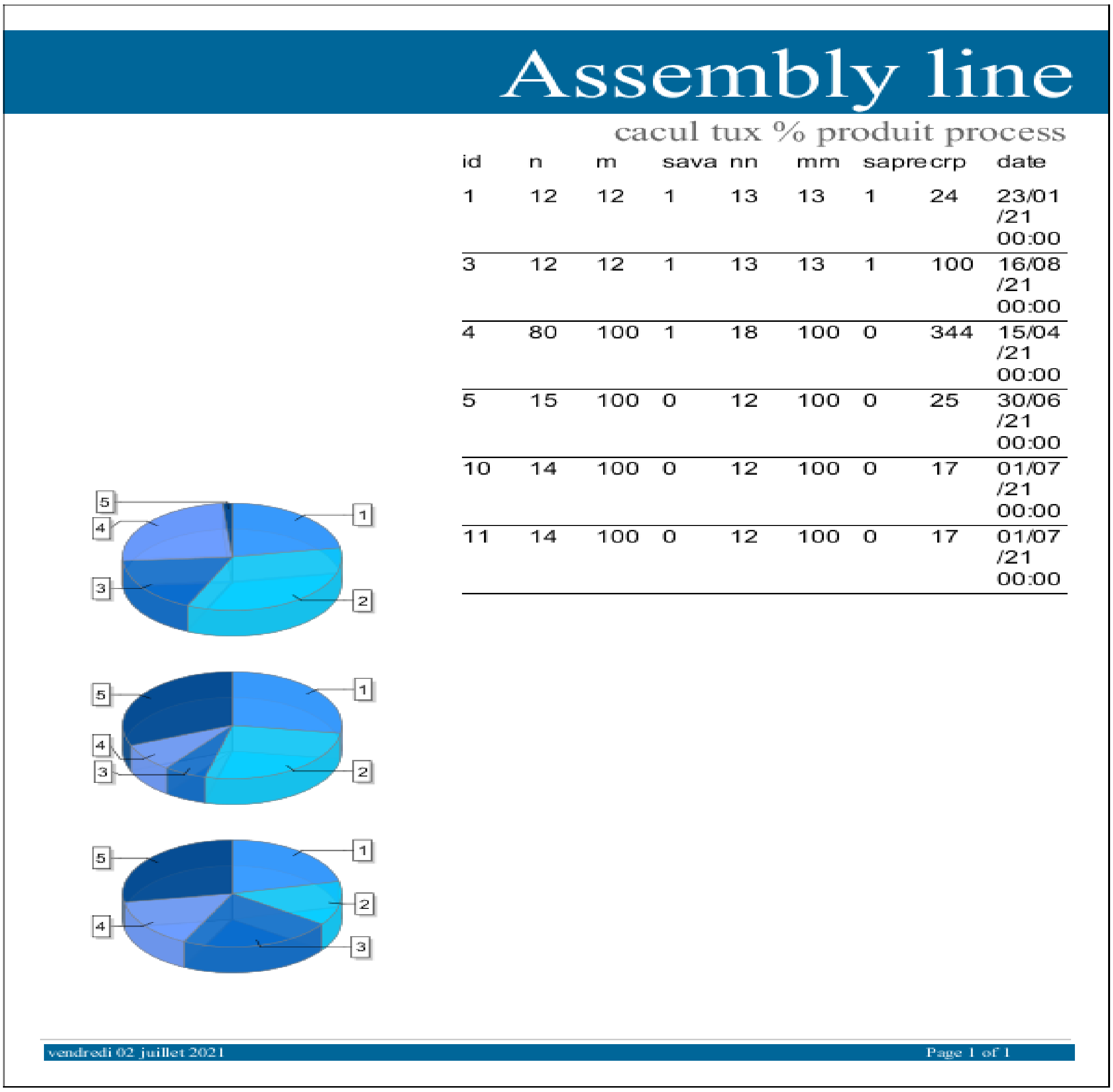
The figure below shows the source code of line chart interface setting:



```
@FXML
void Loadchart() {

    String sql = "select date,Crp from confugiration ORDER BY Crp asc";

    XYChart.Series<String,Double>series=new XYChart.Series<>();
    try{

        conDB();
        ResultSet rs =connection.createStatement().executeQuery(sql);
        while(rs.next()){
            series.getData().add(new XYChart.Data<>(rs.getString( columnLabel: "date"),rs.getDouble( colu
        }

        linechart.getData().add(series);

    }catch (Exception e){

        JOptionPane.showMessageDialog( parentComponent: null, e);
    }
```

Figure (10): The source code of line chart interface setting.

The figure below (Figure 11) presents report editing:



Assembly line

cacul tux % produit process

id	n	m	sava	nn	mm	sapre	crp	date
1	12	12	1	13	13	1	24	23/01/21 00:00
3	12	12	1	13	13	1	100	16/08/21 00:00
4	80	100	1	18	100	0	344	15/04/21 00:00
5	15	100	0	12	100	0	25	30/06/21 00:00
10	14	100	0	12	100	0	17	01/07/21 00:00
11	14	100	0	12	100	0	17	01/07/21 00:00

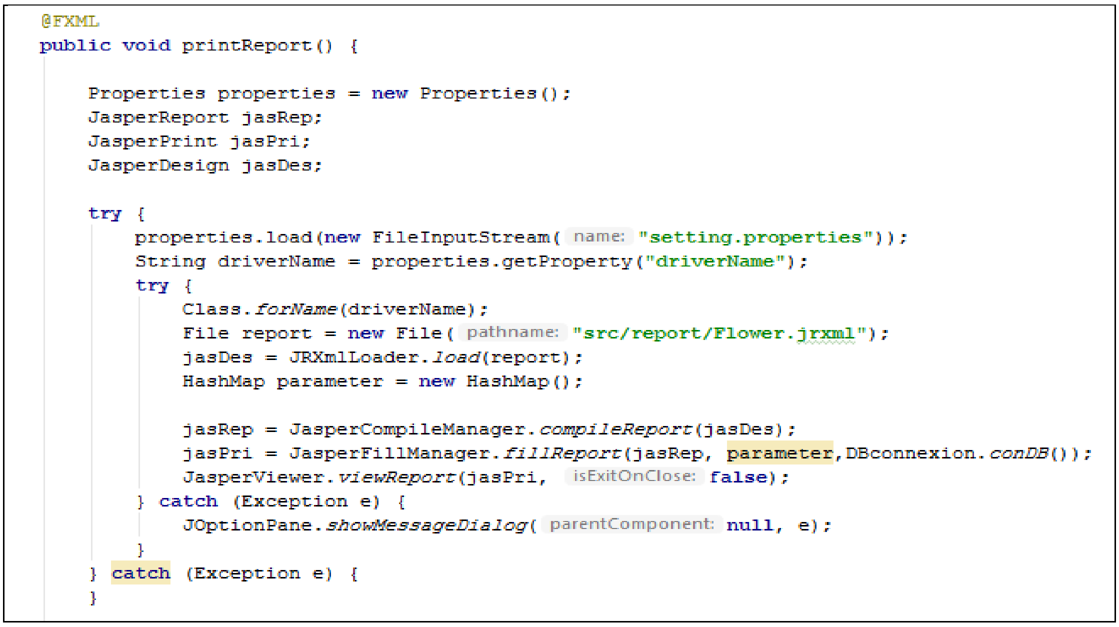
vendredi 02 juillet 2021

Page 1 of 1

Figure (11): *Report of rat production.*

- Conv produits_processes calculate the daily number of production.

The figure below shows the source code of reporting:

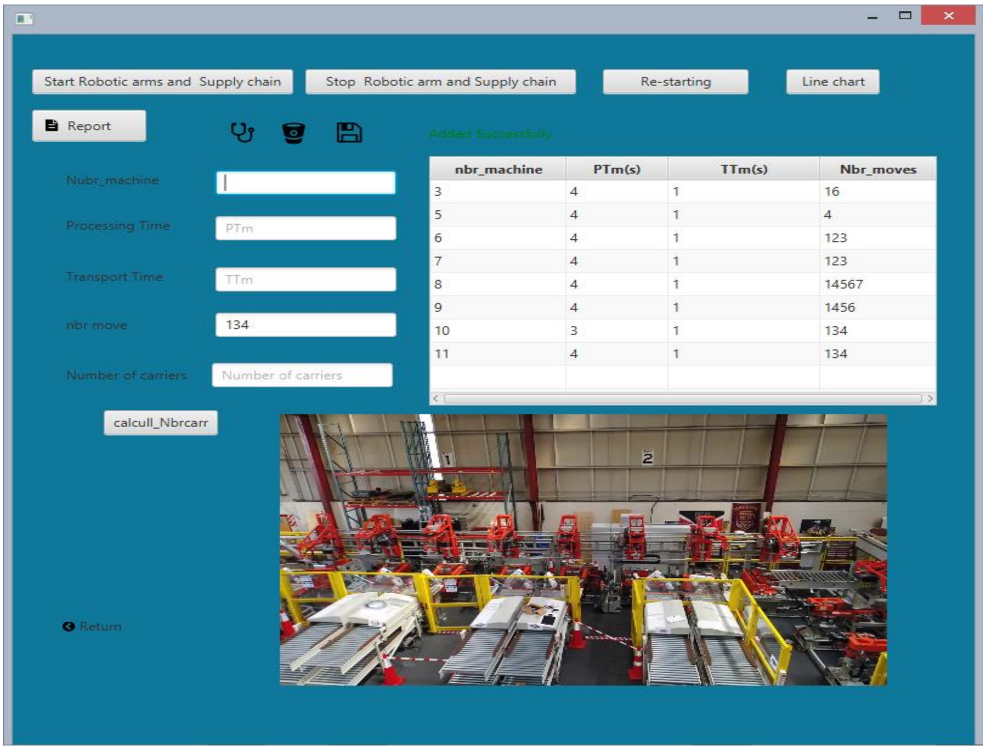


```
@FXML
public void printReport() {

    Properties properties = new Properties();
    JasperReport jasRep;
    JasperPrint jasPri;
    JasperDesign jasDes;

    try {
        properties.load(new FileInputStream( name: "setting.properties"));
        String driverName = properties.getProperty("driverName");
        try {
            Class.forName(driverName);
            File report = new File( pathname: "src/report/Flower.jrxml");
            jasDes = JRXmlLoader.load(report);
            HashMap parameter = new HashMap();

            jasRep = JasperCompileManager.compileReport(jasDes);
            jasPri = JasperFillManager.fillReport(jasRep, parameter,DBconnexion.conDB());
            JasperViewer.viewReport(jasPri,  isExitOnClose: false);
        } catch (Exception e) {
            JOptionPane.showMessageDialog( parentComponent: null, e);
        }
    } catch (Exception e) {
    }
```

Figure (12): *The source code of reporting*

A-Cycle time interface:

The figure below (Figure 13) presents Interface to added cycle time:



Start Robotic arms and Supply chain
Stop Robotic arm and Supply chain
Re-starting
Line chart
Report
calcull_Nbrcarr
Return
1
2

Added Successfully

Nubr_machine

Processing Time: PTm

Transport Time: TTm

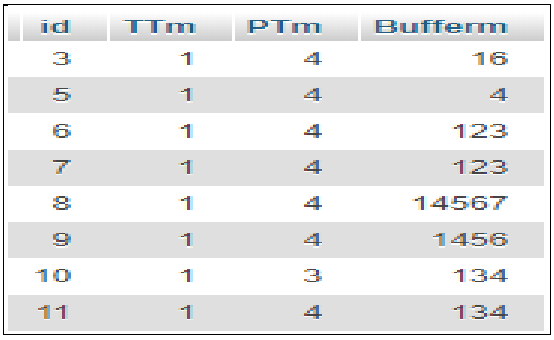
nbr move: 134

Number of carriers: Number of carriers

nbr_machine	PTm(s)	TTm(s)	Nbr_moves
3	4	1	16
5	4	1	4
6	4	1	123
7	4	1	123
8	4	1	14567
9	4	1	1456
10	3	1	134
11	4	1	134

Figure (13): *Interface to added cycle time.*

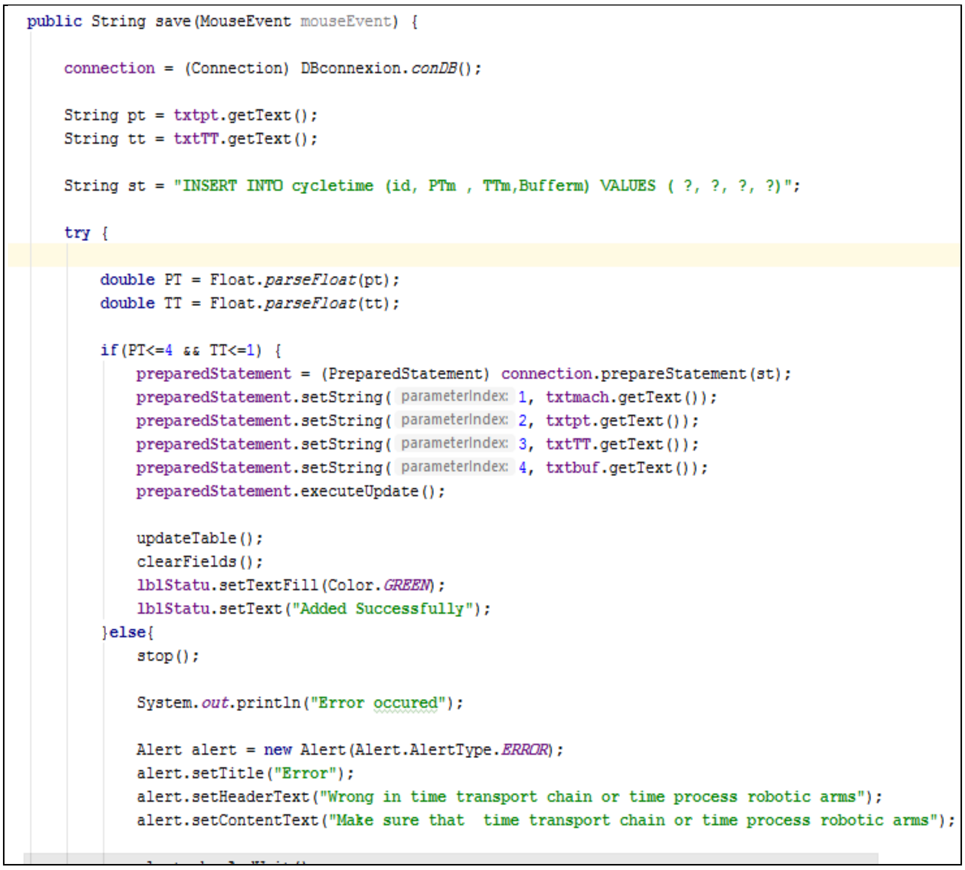
The figure below (Figure 14) presents database of cycle time.



id	TTm	PTm	Bufferm
3	1	4	16
5	1	4	4
6	1	4	123
7	1	4	123
8	1	4	14567
9	1	4	1456
10	1	3	134
11	1	4	134

Figure (14): Database of cycle time.

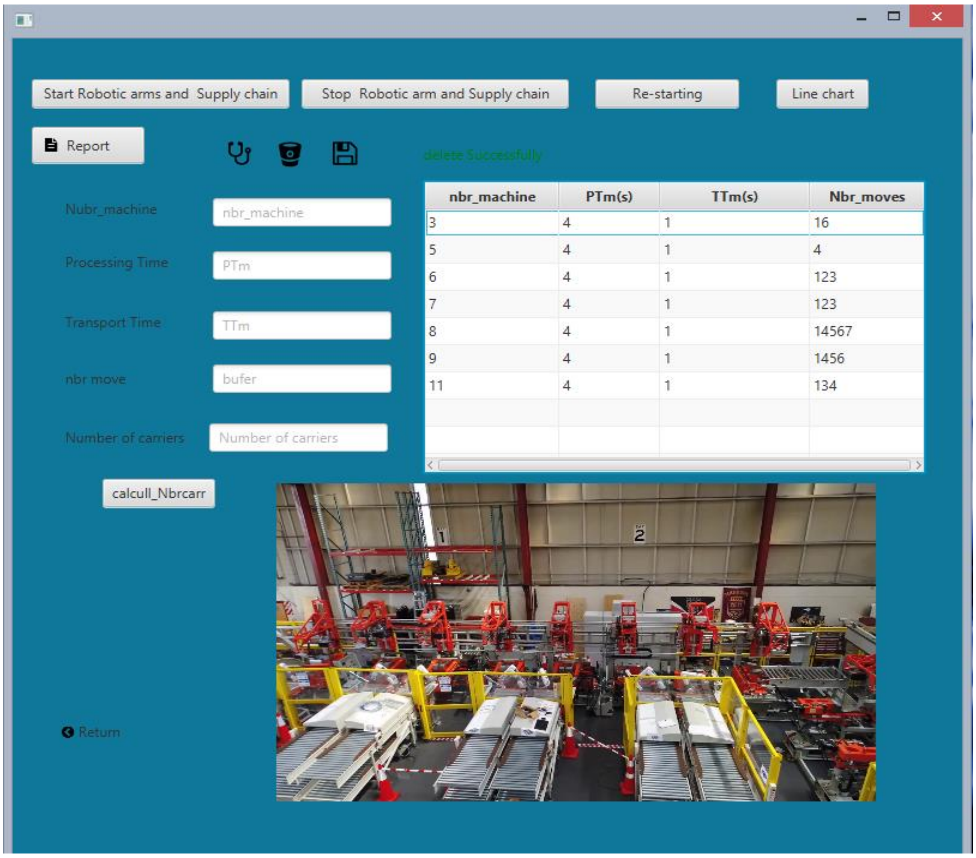
The figure below shows the source code to add by the interface a time cycle:



```
public String save(MouseEvent mouseEvent) {

    connection = (Connection) DBconnexion.conDB();

    String pt = txtpt.getText();
    String tt = txtTT.getText();

    String st = "INSERT INTO cycletime (id, PTm , TTm,Bufferm) VALUES ( ?, ?, ?, ?)";

    try {

        double PT = Float.parseFloat(pt);
        double TT = Float.parseFloat(tt);

        if(PT<=4 && TT<=1) {
            preparedStatement = (PreparedStatement) connection.prepareStatement(st);
            preparedStatement.setString( parameterIndex: 1, txtmach.getText());
            preparedStatement.setString( parameterIndex: 2, txtpt.getText());
            preparedStatement.setString( parameterIndex: 3, txtTT.getText());
            preparedStatement.setString( parameterIndex: 4, txtbuf.getText());
            preparedStatement.executeUpdate();

            updateTable();
            clearFields();
            lblStatu.setTextFill(Color.GREEN);
            lblStatu.setText("Added Successfully");
        }else{
            stop();

            System.out.println("Error occured");

            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error");
            alert.setHeaderText("Wrong in time transport chain or time process robotic arms");
            alert.setContentText("Make sure that  time transport chain or time process robotic arms");
```

Figure (15): The source code to add a cycle time.

The figure below (Figure 16) presents the Interface of cycle time deleted:

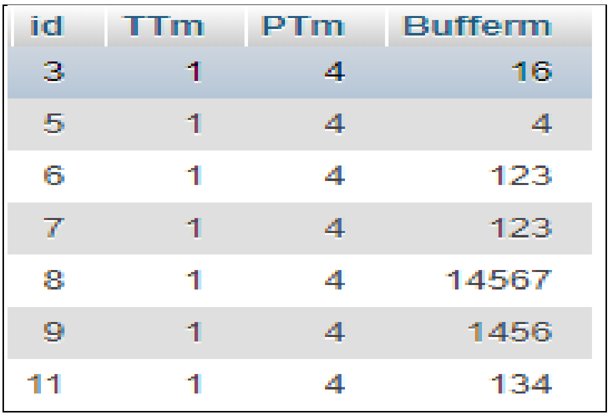


delete Successfully

nbr_machine	PTm(s)	TTm(s)	Nbr_moves
3	4	1	16
5	4	1	4
6	4	1	123
7	4	1	123
8	4	1	14567
9	4	1	1456
11	4	1	134

Figure (16): *Interface cycle time delete.*

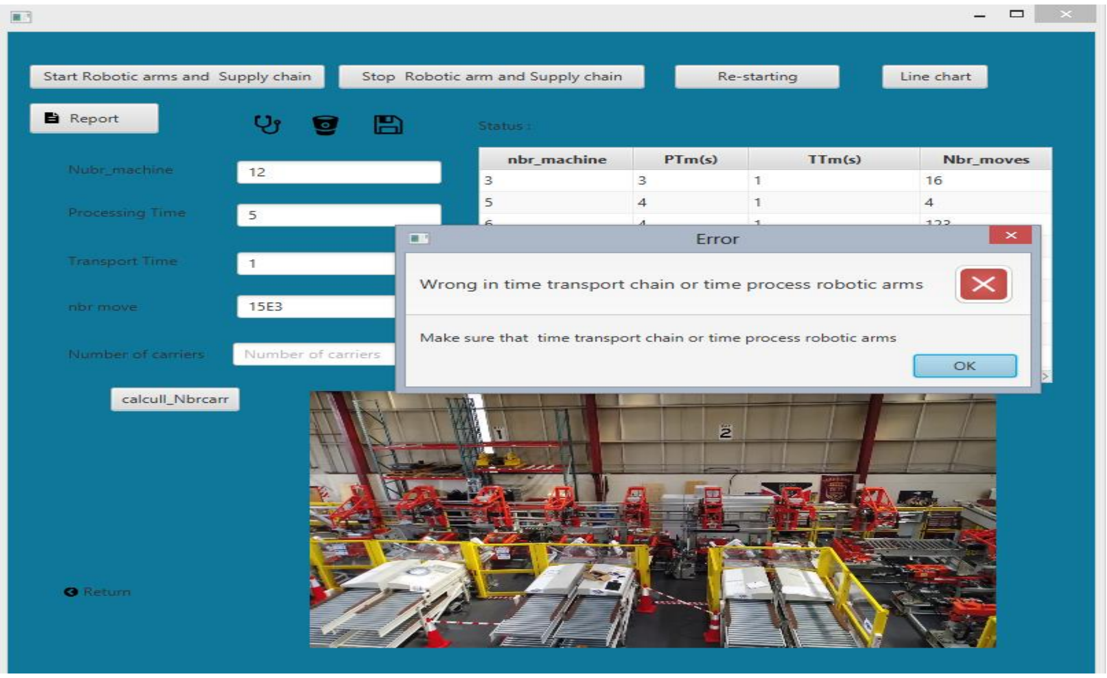
The figure below (Figure 17) presents database to delete a cycle time.



id	TTm	PTm	Bufferm
3	1	4	16
5	1	4	4
6	1	4	123
7	1	4	123
8	1	4	14567
9	1	4	1456
11	1	4	134

Figure (17): Database to delete a cycle time.

The figure below (Figure 18) presents Interface cycle time when a problem appears:


Start Robotic arms and Supply chain
Stop Robotic arm and Supply chain
Re-starting
Line chart
Report
Status :
calcull_Nbrcarr
Return
1
2

Nubr_machine: 12

Processing Time: 5

Transport Time: 1

nbr move: 15E3

Number of carriers: Number of carriers

nbr_machine	PTm(s)	TTm(s)	Nbr_moves
3	3	1	16
5	4	1	4
6	4	1	123

Error

Wrong in time transport chain or time process robotic arms

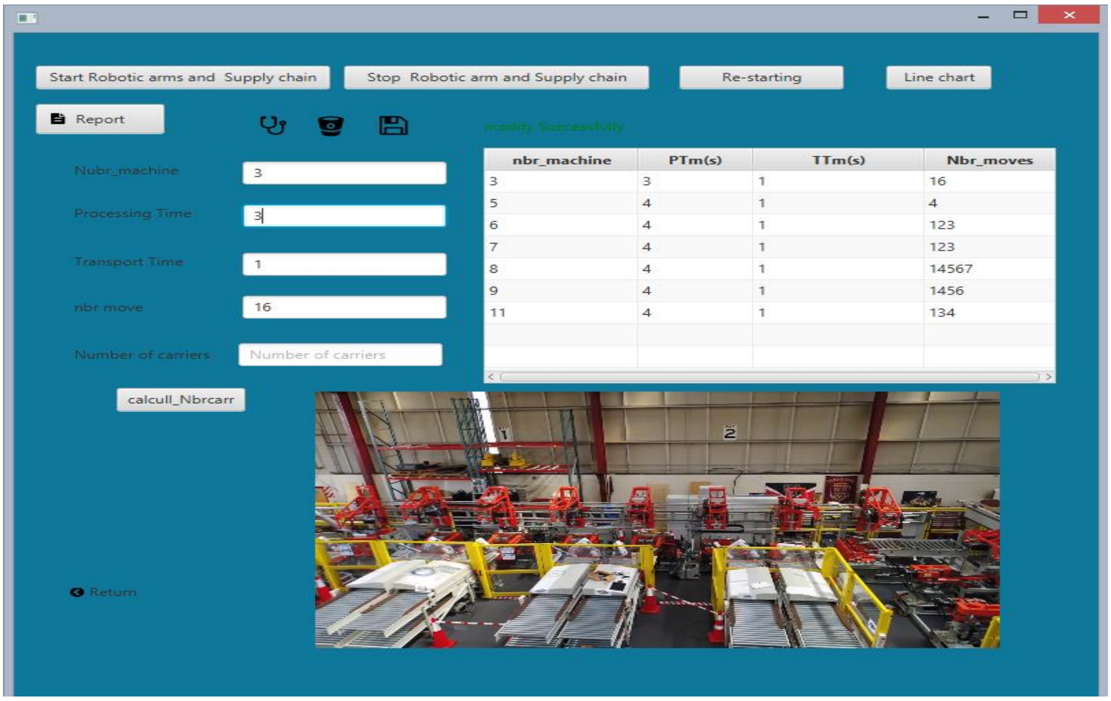
Make sure that time transport chain or time process robotic arms

OK

Figure (18): *Interface of cycle time when a problem appears.*

-The cycle time assembly line of cars need 4 min or less to work robotic arms (production time) and 1 min or less to work supply chain (transport time) , if the production time transcend 4min or transport time transcend 1min , the system block and stopped the video of the application,

The figure below (Figure 19) presents Interface to modify cycle time:


Start Robotic arms and Supply chain
Stop Robotic arm and Supply chain
Re-starting
Line chart
Report
calcull_Nbrcarr
Return
1
2

modify Successfully

Nubr_machine: 3

Processing Time: 3

Transport Time: 1

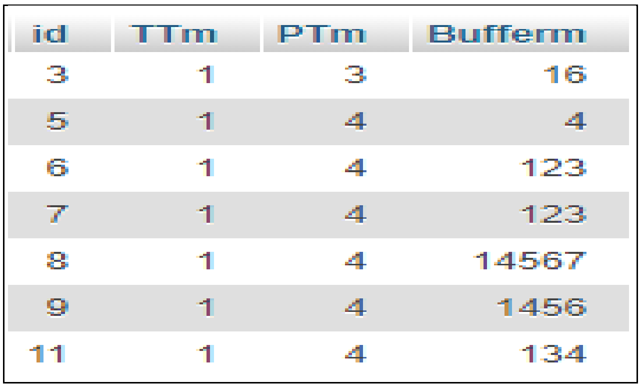
nbr move: 16

Number of carriers: Number of carriers

nbr_machine	PTm(s)	TTm(s)	Nbr_moves
3	3	1	16
5	4	1	4
6	4	1	123
7	4	1	123
8	4	1	14567
9	4	1	1456
11	4	1	134

Figure (19): *Interface of cycle time modification.*

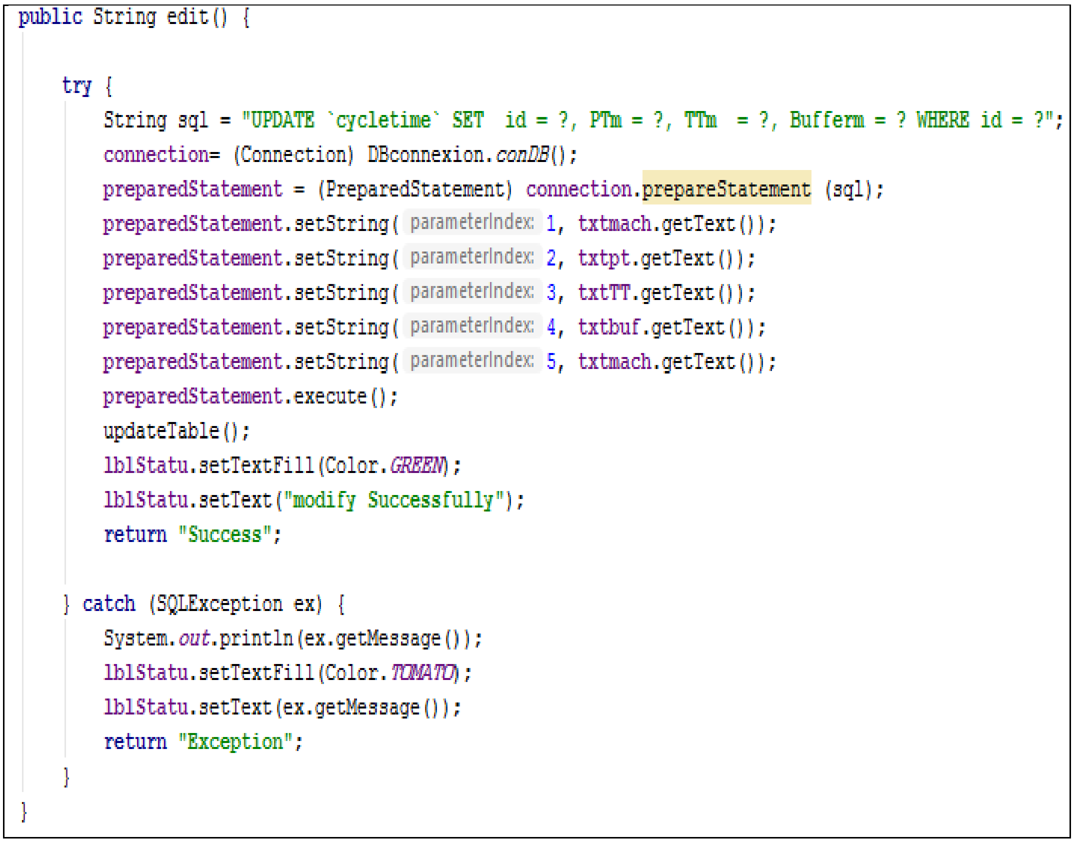
The figure below (Figure 20) presents the database of cycle time modification.



id	TTm	PTm	Bufferm
3	1	3	16
5	1	4	4
6	1	4	123
7	1	4	123
8	1	4	14567
9	1	4	1456
11	1	4	134

Figure (20): *Database of cycle time modification.*

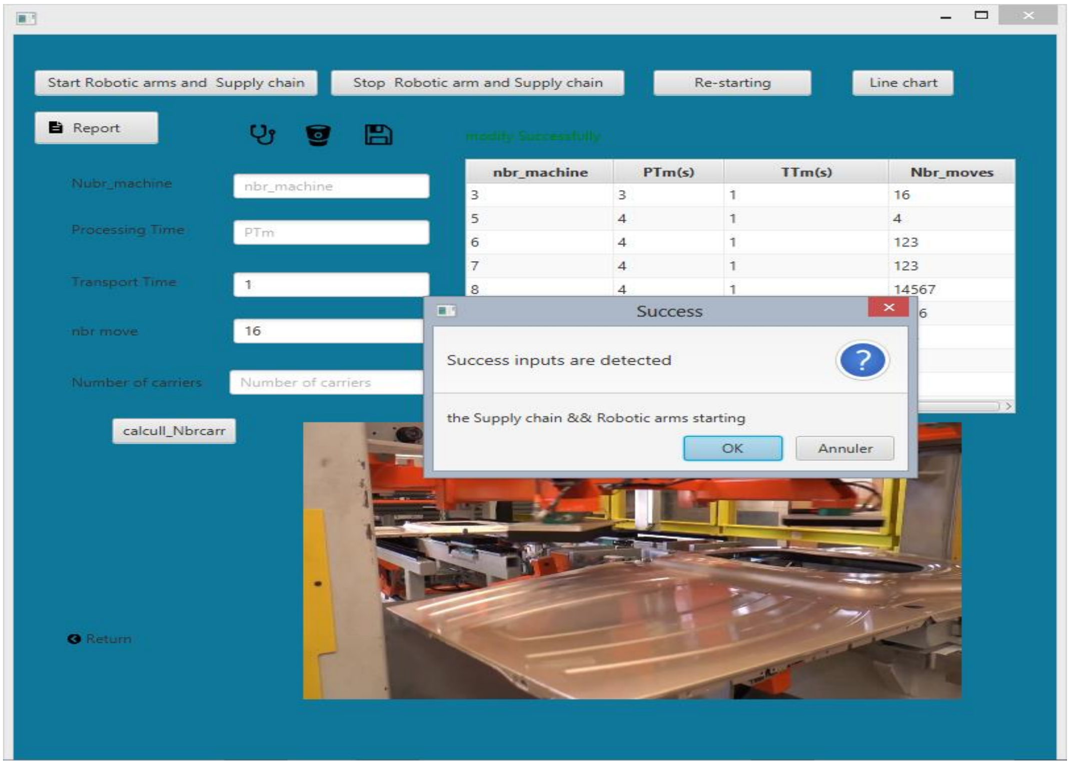
The figure below shows the source code of cycle time modification:



```
public String edit() {

    try {
        String sql = "UPDATE `cycletime` SET  id = ?, PTm = ?, TTm  = ?, Bufferm = ? WHERE id = ?";
        connection= (Connection) DBconnexion.conDB();
        preparedStatement = (PreparedStatement) connection.prepareStatement (sql);
        preparedStatement.setString( parameterIndex: 1, txtmach.getText());
        preparedStatement.setString( parameterIndex: 2, txtpt.getText());
        preparedStatement.setString( parameterIndex: 3, txtTT.getText());
        preparedStatement.setString( parameterIndex: 4, txtbuf.getText());
        preparedStatement.setString( parameterIndex: 5, txtmach.getText());
        preparedStatement.execute();
        updateTable();
        lblStatu.setTextFill(Color.GREEN);
        lblStatu.setText("modify Successfully");
        return "Success";

    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
        lblStatu.setTextFill(Color.TOMATO);
        lblStatu.setText(ex.getMessage());
        return "Exception";
    }
}
```

Figure (21): *The source code of modification cycle time.*

The figure below (Figure 22) presents the supply chain and robotic arms starting:


Start Robotic arms and Supply chain
Stop Robotic arm and Supply chain
Re-starting
Line chart
Report
modify Successfully
calcull_Nbrcarr
Return

Nubr_machine: nbr_machine

Processing Time: PTm

Transport Time: 1

nbr move: 16

Number of carriers: Number of carriers

nbr_machine	PTm(s)	TTm(s)	Nbr_moves
3	3	1	16
5	4	1	4
6	4	1	123
7	4	1	123
8	4	1	14567
			6

Success

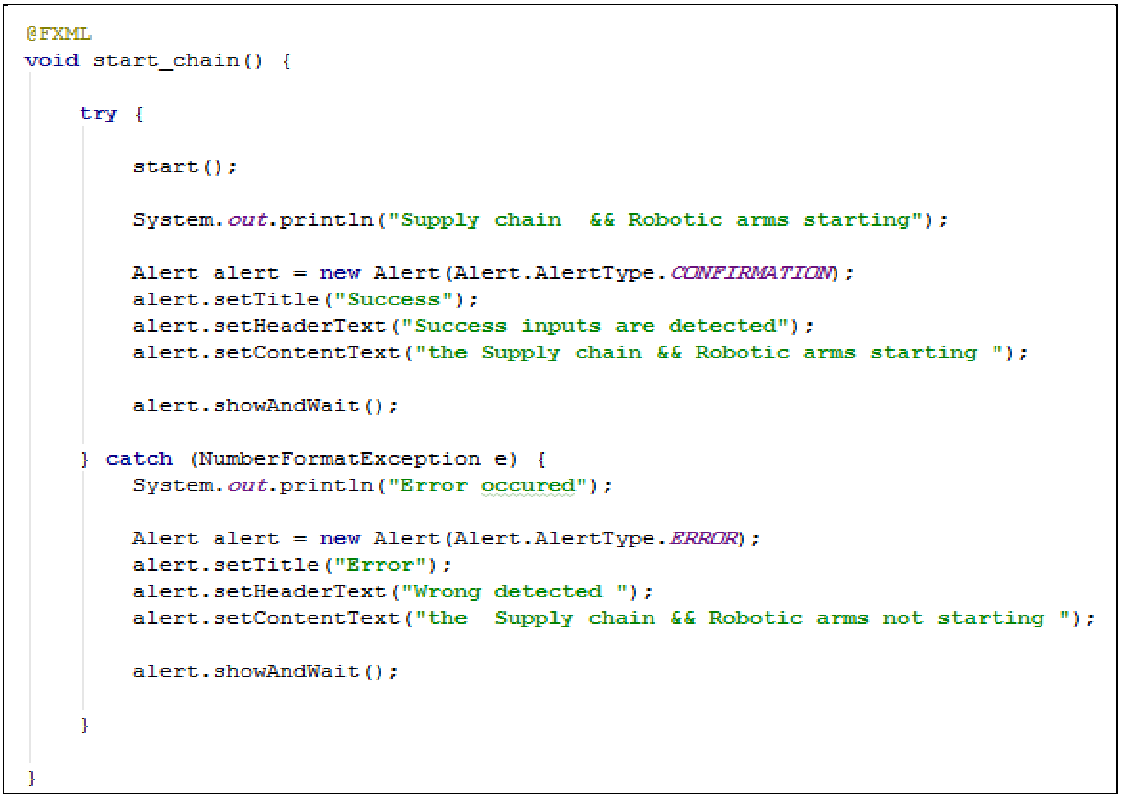
Success inputs are detected

the Supply chain && Robotic arms starting

OK   Annuler

Figure (22): *The supply chain and robotic arms starting.*

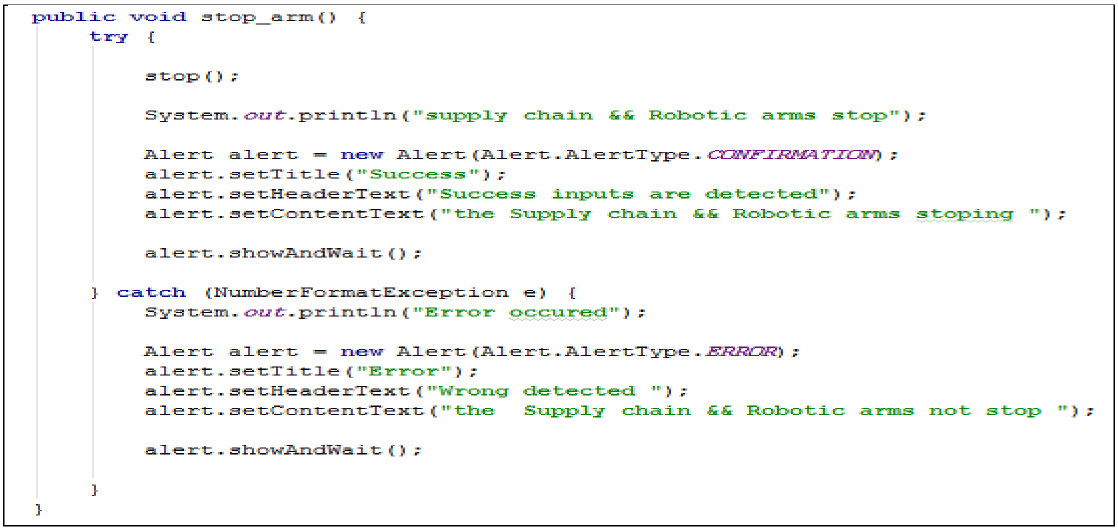
The figure below shows the source code of the supply chain and robotic arms starting in interface cycle time:



```
@FXML
void start_chain() {

    try {

        start();

        System.out.println("Supply chain  && Robotic arms starting");

        Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
        alert.setTitle("Success");
        alert.setHeaderText("Success inputs are detected");
        alert.setContentText("the Supply chain && Robotic arms starting ");

        alert.showAndWait();

    } catch (NumberFormatException e) {
        System.out.println("Error occured");

        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error");
        alert.setHeaderText("Wrong detected ");
        alert.setContentText("the  Supply chain && Robotic arms not starting ");

        alert.showAndWait();

    }

}
```

Figure (23): *The supply chain and robotic arms starting.*

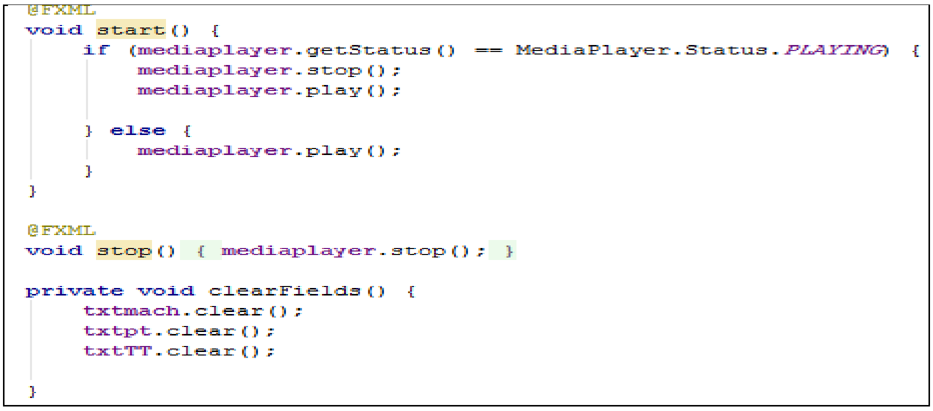
The figure below shows the source code to stop supply chain and robotic arms:



```
public void stop_arm() {
        try {

            stop();

            System.out.println("supply chain && Robotic arms stop");

            Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
            alert.setTitle("Success");
            alert.setHeaderText("Success inputs are detected");
            alert.setContentText("the Supply chain && Robotic arms stoping ");

            alert.showAndWait();

        } catch (NumberFormatException e) {
            System.out.println("Error occured");

            Alert alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error");
            alert.setHeaderText("Wrong detected ");
            alert.setContentText("the  Supply chain && Robotic arms not stop ");

            alert.showAndWait();

        }
}
```

Figure (24): *The supply chain and robotic arms stopping.*

The figure below shows the source code to start and stop video:



```
@FXML
void start() {
    if (mediaplayer.getStatus() == MediaPlayer.Status.PLAYING) {
        mediaplayer.stop();
        mediaplayer.play();

    } else {
        mediaplayer.play();
    }
}

@FXML
void stop() { mediaplayer.stop(); }

private void clearFields() {
    txtmach.clear();
    txtpt.clear();
    txtTT.clear();

}
```

Figure (25): *The source code start and stop video.*

The figure below (Figure 26) presents Interface line chart of cycle time:

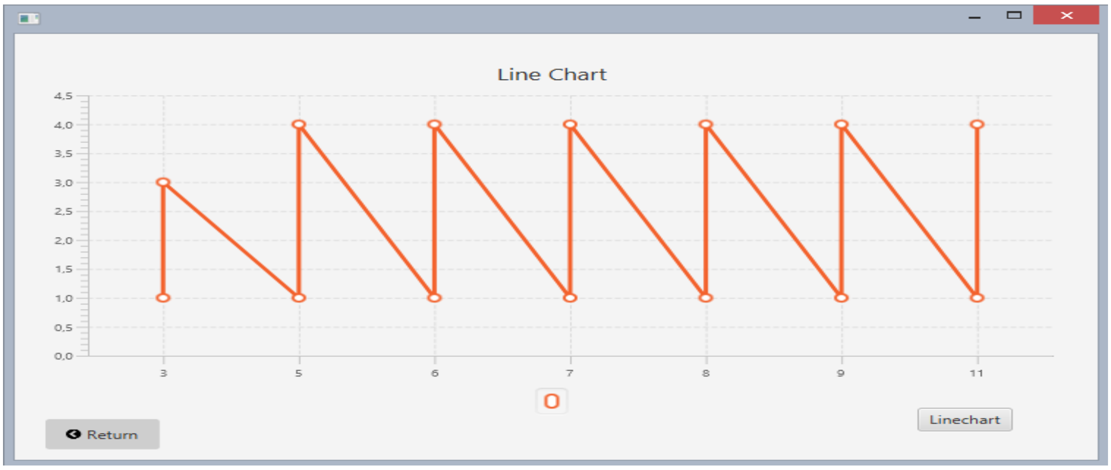
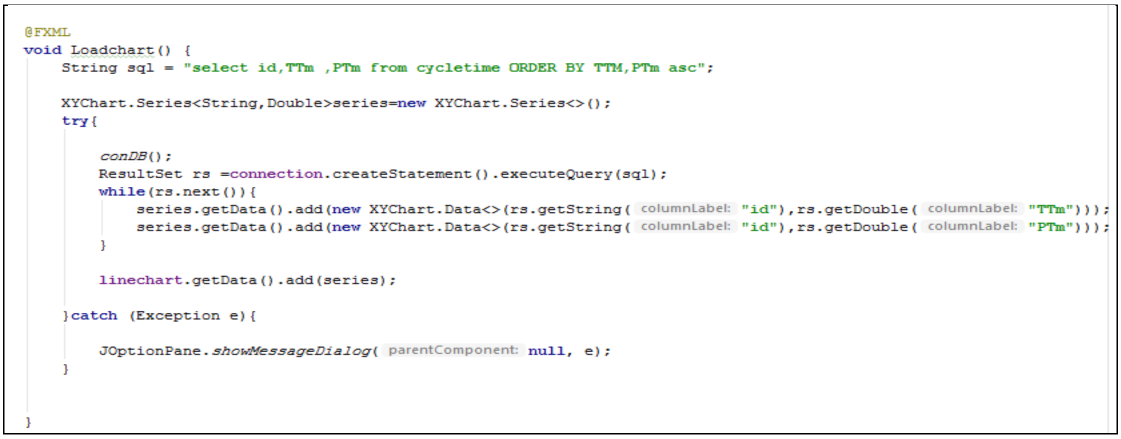


Figure (26): *Interface line chart of cycle time.*

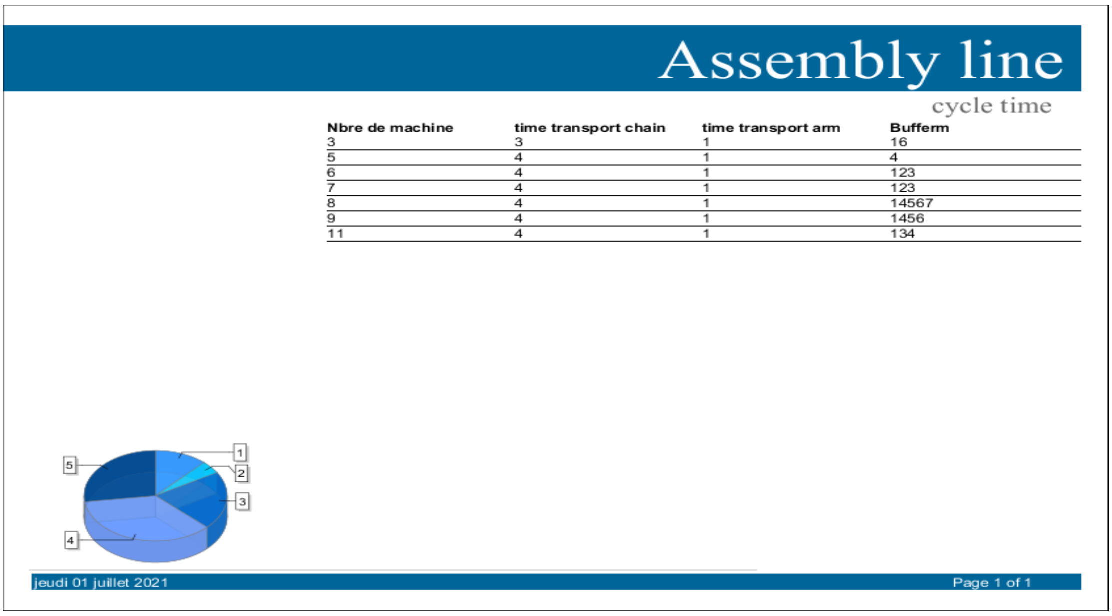
The figure below shows the source code line chart of cycle time:



```
@FXML
void Loadchart() {
    String sql = "select id,TTm ,PTm from cycletime ORDER BY TTM,PTm asc";

    XYChart.Series<String,Double>series=new XYChart.Series<>();
    try{

        conDB();
        ResultSet rs =connection.createStatement().executeQuery(sql);
        while(rs.next()){
            series.getData().add(new XYChart.Data<>(rs.getString( columnLabel: "id"),rs.getDouble( columnLabel: "TTm")));
            series.getData().add(new XYChart.Data<>(rs.getString( columnLabel: "id"),rs.getDouble( columnLabel: "PTm")));
        }

        linechart.getData().add(series);

    }catch (Exception e){

        JOptionPane.showMessageDialog( parentComponent: null, e);
    }


}
```

Figure (27): *The source code of line chart in cycle time.*

The figure below (Figure 28) presents the reporting interface:



Assembly line

cycle time

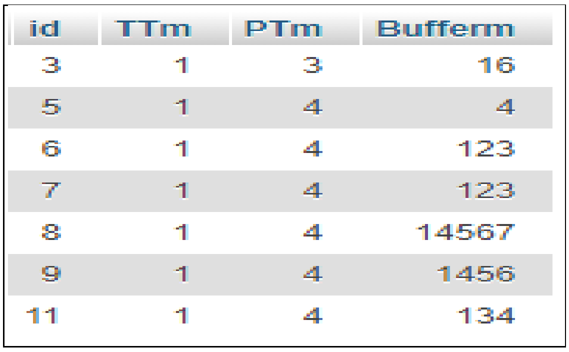
Nbre de machine	time transport chain	time transport arm	Bufferm
3	3	1	16
5	4	1	4
6	4	1	123
7	4	1	123
8	4	1	14567
9	4	1	1456
11	4	1	134

jeudi 01 juillet 2021

Page 1 of 1

Figure (28): *Reporting in a cycle time.*

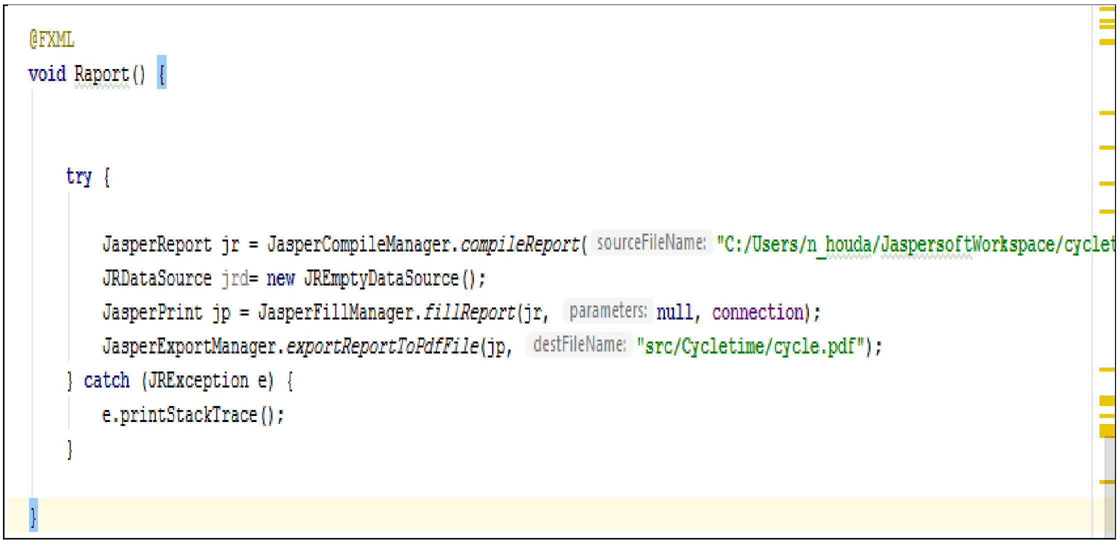
The figure below (Figure 29) presents database of cycle time of a report.



id	TTm	PTm	Bufferm
3	1	3	16
5	1	4	4
6	1	4	123
7	1	4	123
8	1	4	14567
9	1	4	1456
11	1	4	134

Figure (29): *Database of a report in cycle time.*

The figure below shows the source code of the interface that added cycle time of report:

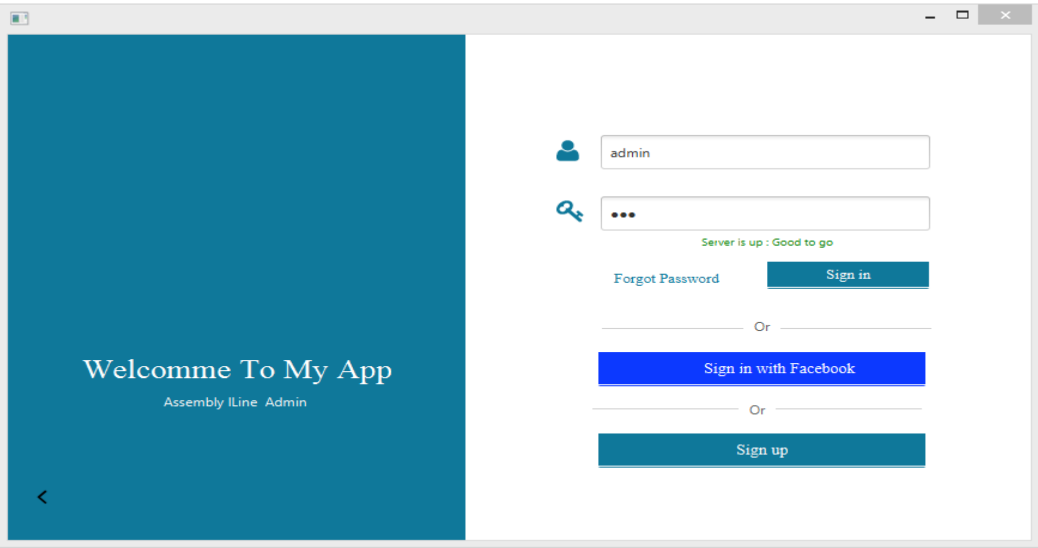


```
@FXML
void Raport() {


    try {

        JasperReport jr = JasperCompileManager.compileReport( sourceFileName: "C:/Users/n_houda/JaspersoftWorkspace/cyclet
        JRDataSource jrd= new JREmptyDataSource();
        JasperPrint jp = JasperFillManager.fillReport(jr,  parameters: null, connection);
        JasperExportManager.exportReportToPdfFile(jp,  destFileName: "src/Cycletime/cycle.pdf");
    } catch (JRException e) {
        e.printStackTrace();
    }

}
```

Figure (30): *The source code of the interface that added cycle time of report.*

3. The authentication admin interface

The figure below (Figure 31) presents the authentication admin interface:



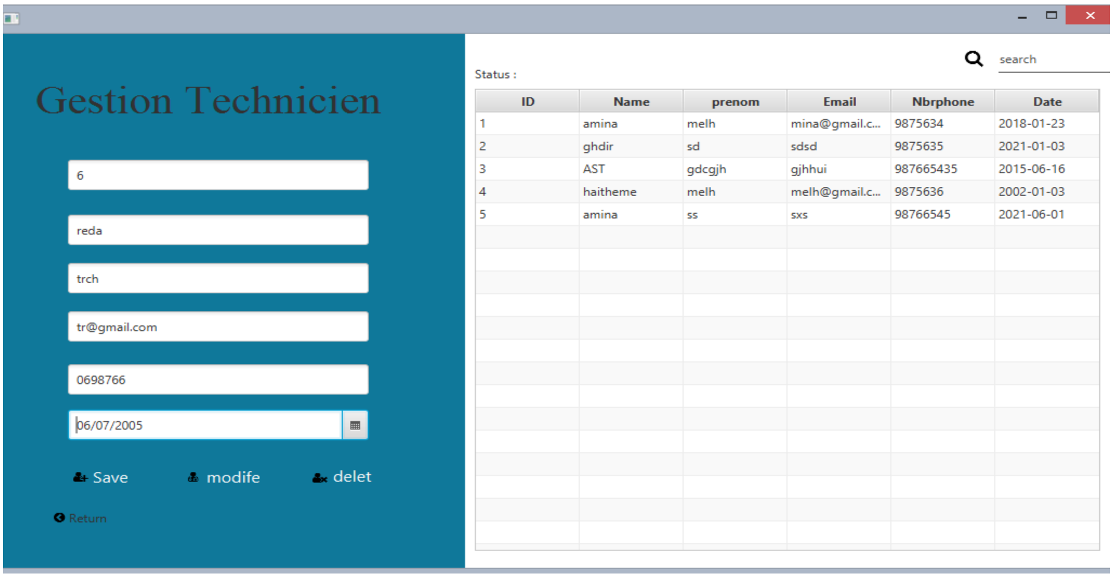
admin

•••

Server is up : Good to go

Figure 31: *The authentication admin interface.*

The figure below (Figure 32) presents the interface to manage technician count:



Gestion Technicien

6

reda

trch

tr@gmail.com

0698766

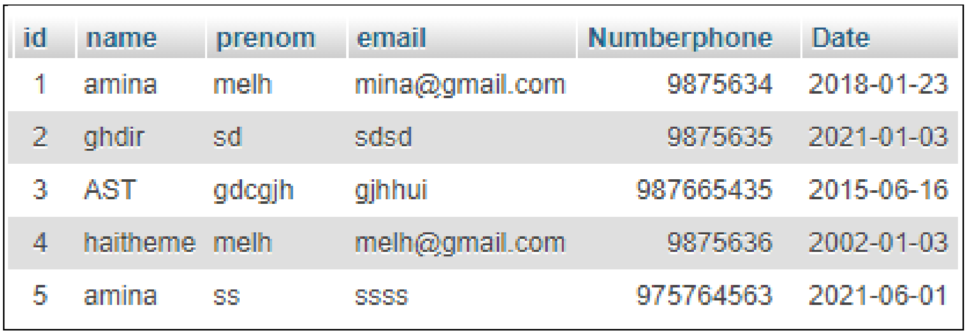
06/07/2005

Status :

ID	Name	prenom	Email	Nbrphone	Date
1	amina	melh	mina@gmail.c...	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
3	AST	gdcgjh	gjhhui	987665435	2015-06-16
4	haitheme	melh	melh@gmail.c...	9875636	2002-01-03
5	amina	ss	sxs	98766545	2021-06-01

Figure 32: *The interface management technician count.*

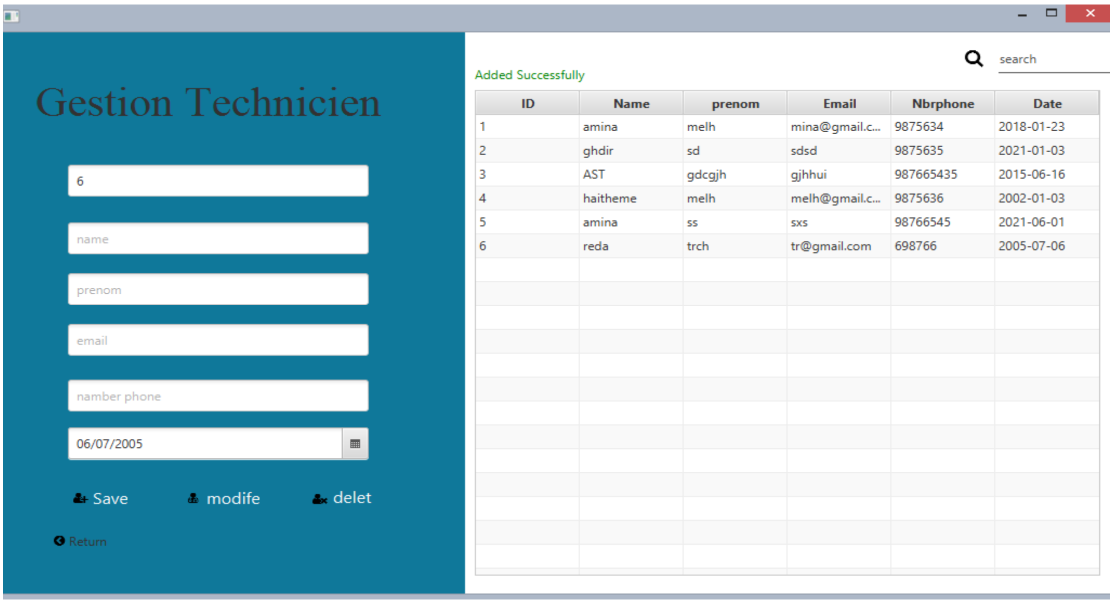
The figure below (Figure 33) presents database of technician management.



id	name	prenom	email	Numberphone	Date
1	amina	melh	mina@gmail.com	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
3	AST	gdcgjh	gjhhui	987665435	2015-06-16
4	haitheme	melh	melh@gmail.com	9875636	2002-01-03
5	amina	ss	ssss	975764563	2021-06-01

Figure 33: *The interface technician management.*

The figure below (Figure 34) presents the interface to adding a new technician:



Gestion Technicien

6

name

prenom

email

namber phone

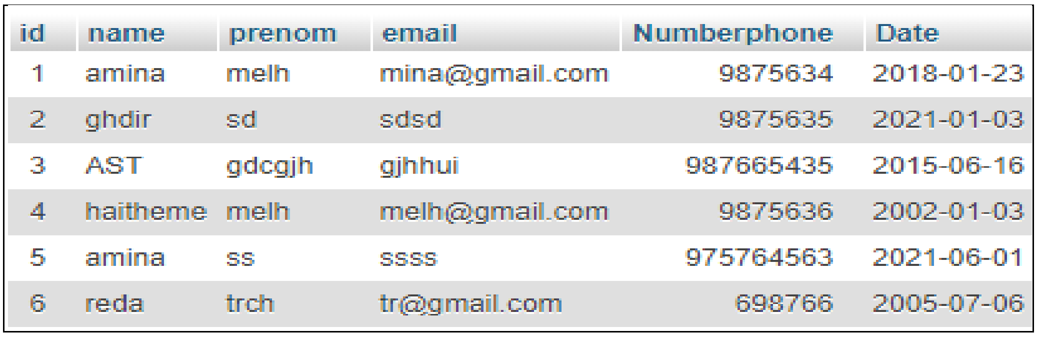
06/07/2005

Added Successfully

ID	Name	prenom	Email	Nbrphone	Date
1	amina	melh	mina@gmail.c...	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
3	AST	gdcgjh	gjhhui	987665435	2015-06-16
4	haitheme	melh	melh@gmail.c...	9875636	2002-01-03
5	amina	ss	sxs	98766545	2021-06-01
6	reda	trch	tr@gmail.com	698766	2005-07-06

Figure (34): *The interface to adding a new technician.*

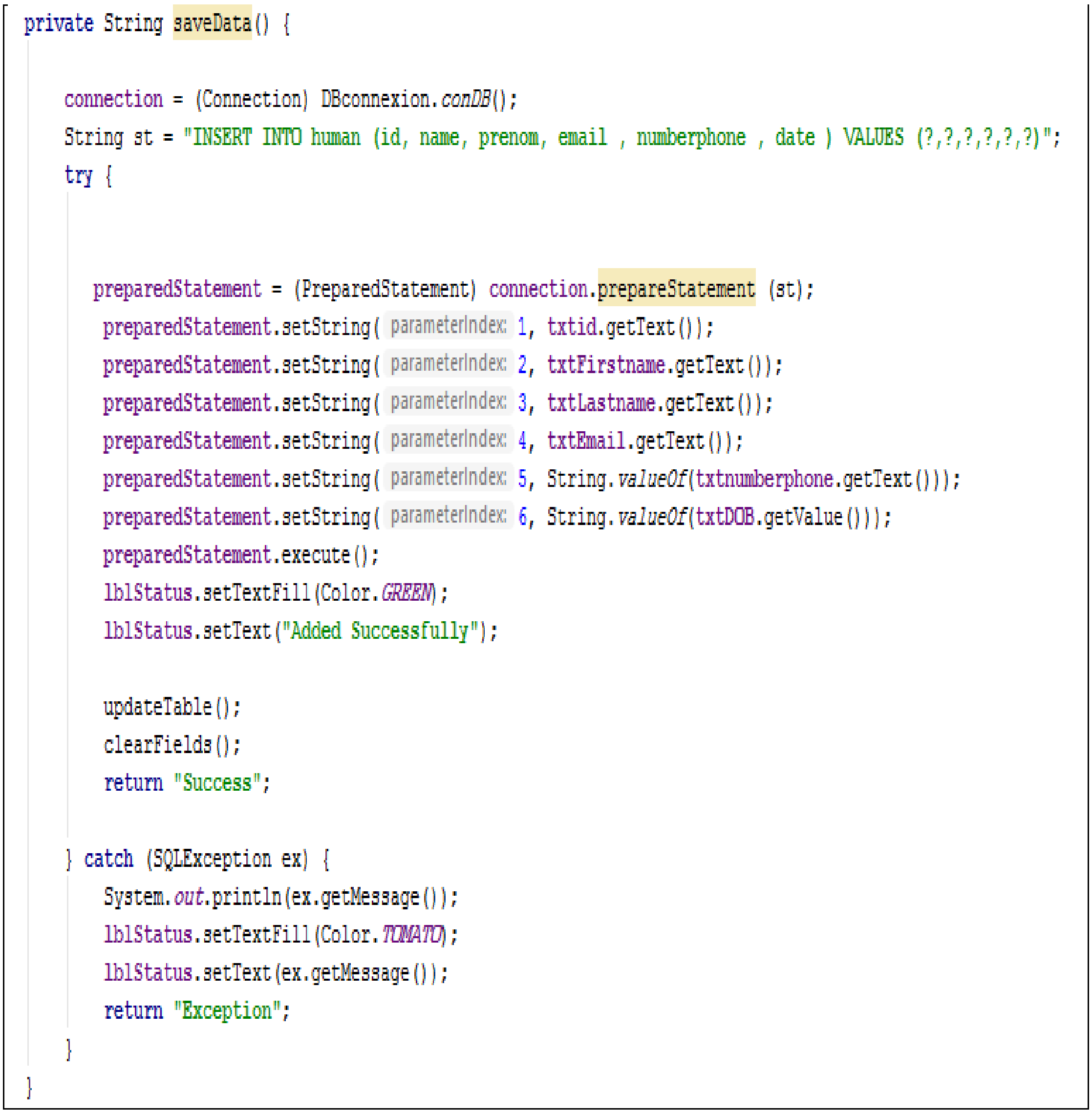
The figure below (Figure 35) presents the database adding a new technician.



id	name	prenom	email	Numberphone	Date
1	amina	melh	mina@gmail.com	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
3	AST	gdcgjh	gjhhui	987665435	2015-06-16
4	haitheme	melh	melh@gmail.com	9875636	2002-01-03
5	amina	ss	ssss	975764563	2021-06-01
6	reda	trch	tr@gmail.com	698766	2005-07-06

Figure (35): *Adding in database of a new technician.*

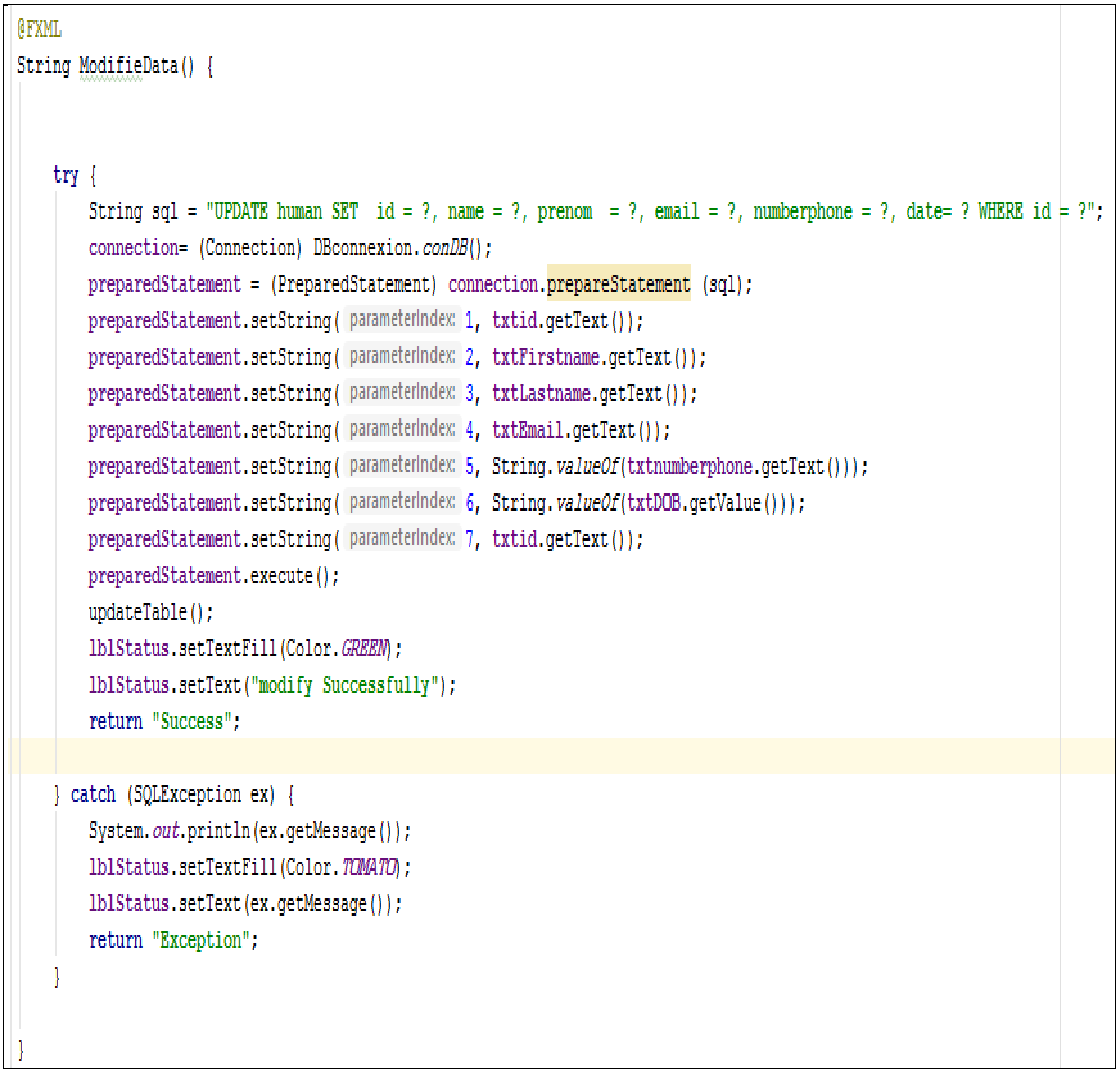
The figure below shows the source code to add a new technician:



```
private String saveData() {

    connection = (Connection) DBconnexion.conDB();
    String st = "INSERT INTO human (id, name, prenom, email , numberphone , date ) VALUES (?,?,?,?,?,?)";
    try {


       preparedStatement = (PreparedStatement) connection.prepareStatement (st);
        preparedStatement.setString( parameterIndex: 1, txtid.getText());
        preparedStatement.setString( parameterIndex: 2, txtFirstname.getText());
        preparedStatement.setString( parameterIndex: 3, txtLastname.getText());
        preparedStatement.setString( parameterIndex: 4, txtEmail.getText());
        preparedStatement.setString( parameterIndex: 5, String.valueOf(txtnumberphone.getText()));
        preparedStatement.setString( parameterIndex: 6, String.valueOf(txtDOB.getValue()));
        preparedStatement.execute();
        lblStatus.setTextFill(Color.GREEN);
        lblStatus.setText("Added Successfully");

        updateTable();
        clearFields();
        return "Success";

    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
        lblStatus.setTextFill(Color.TOMATO);
        lblStatus.setText(ex.getMessage());
        return "Exception";
    }
}
```

Figure (36): *The source code to add a new technician.*

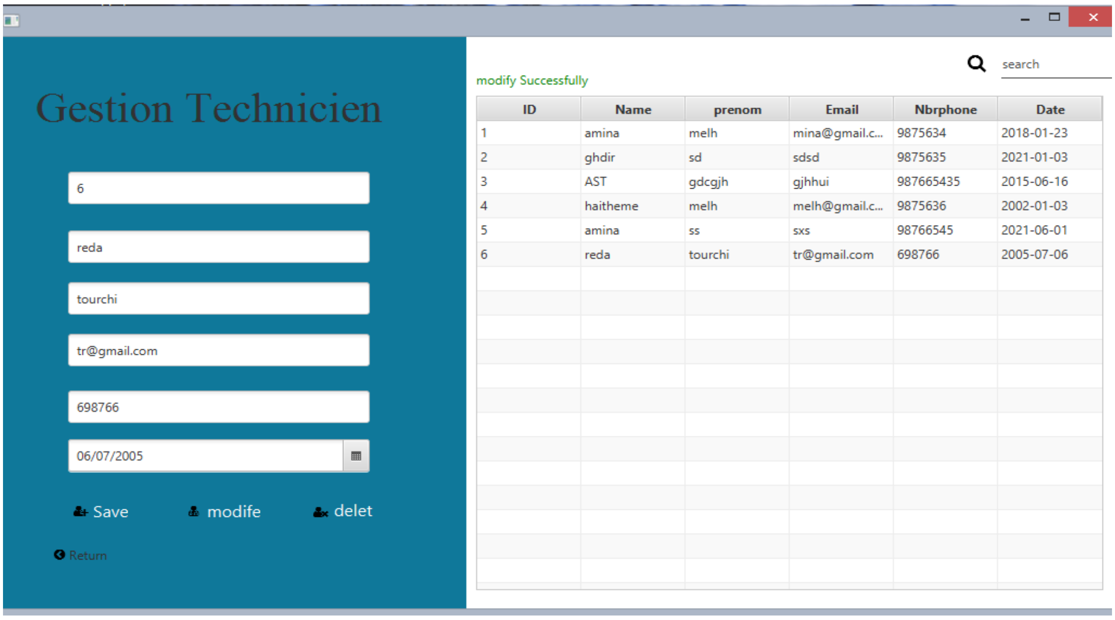
The figure below shows the source code to modify technician information:



```
@FXML
String ModifieData() {


    try {
        String sql = "UPDATE human SET  id = ?, name = ?, prenom  = ?, email = ?, numberphone = ?, date= ? WHERE id = ?";
        connection= (Connection) DBconnexion.conDB();
        preparedStatement = (PreparedStatement) connection.prepareStatement (sql);
        preparedStatement.setString( parameterIndex: 1, txtid.getText());
        preparedStatement.setString( parameterIndex: 2, txtFirstname.getText());
        preparedStatement.setString( parameterIndex: 3, txtLastname.getText());
        preparedStatement.setString( parameterIndex: 4, txtEmail.getText());
        preparedStatement.setString( parameterIndex: 5, String.valueOf(txtnumberphone.getText()));
        preparedStatement.setString( parameterIndex: 6, String.valueOf(txtDOB.getValue()));
        preparedStatement.setString( parameterIndex: 7, txtid.getText());
        preparedStatement.execute();
        updateTable();
        lblStatus.setTextFill(Color.GREEN);
        lblStatus.setText("modify Successfully");
        return "Success";

    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
        lblStatus.setTextFill(Color.TOMATO);
        lblStatus.setText(ex.getMessage());
        return "Exception";
    }

}
```

Figure (37): *The source code to modify technician information.*

The figure below (Figure 38) presents the interface technician modification:



modify Successfully

6

reda

tourchi

tr@gmail.com

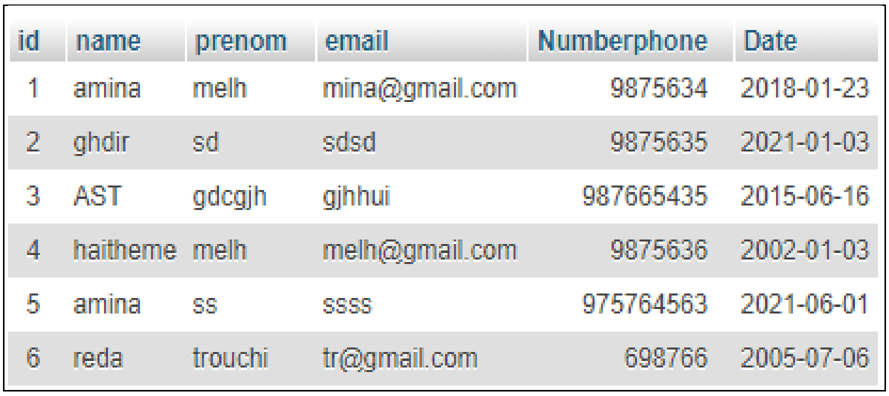
698766

06/07/2005

ID	Name	prenom	Email	Nbrphone	Date
1	amina	melh	mina@gmail.c...	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
3	AST	gdcgjh	gjhhui	987665435	2015-06-16
4	haitheme	melh	melh@gmail.c...	9875636	2002-01-03
5	amina	ss	sxs	98766545	2021-06-01
6	reda	tourchi	tr@gmail.com	698766	2005-07-06

Figure 38: *The admin modify the technician count.*

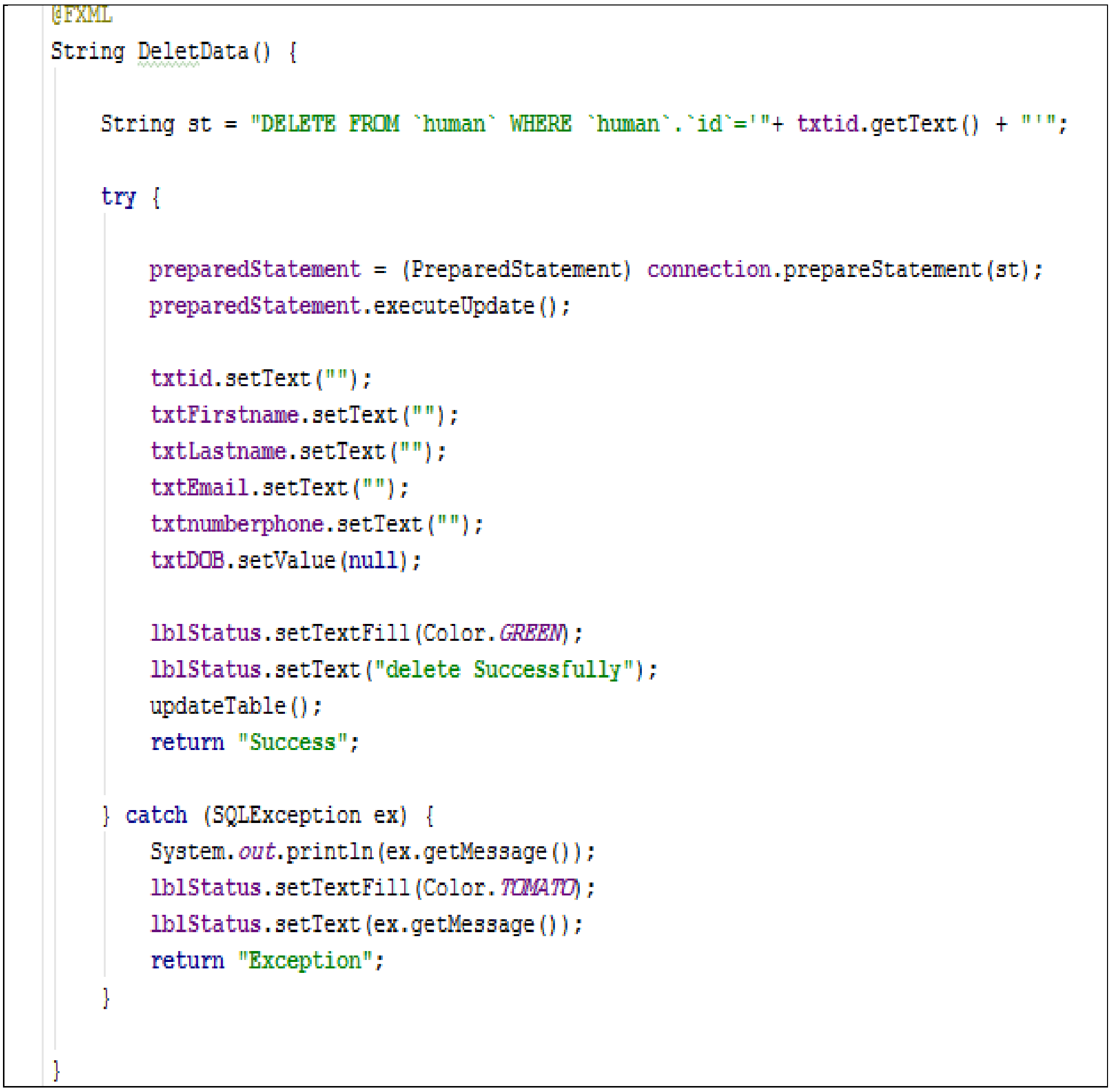
The figure below (Figure 39) presents technician data base modification.



id	name	prenom	email	Numberphone	Date
1	amina	melh	mina@gmail.com	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
3	AST	gdcgjh	gjhhui	987665435	2015-06-16
4	haitheme	melh	melh@gmail.com	9875636	2002-01-03
5	amina	ss	ssss	975764563	2021-06-01
6	reda	trouchi	tr@gmail.com	698766	2005-07-06

Figure 39: *The database technician modification.*

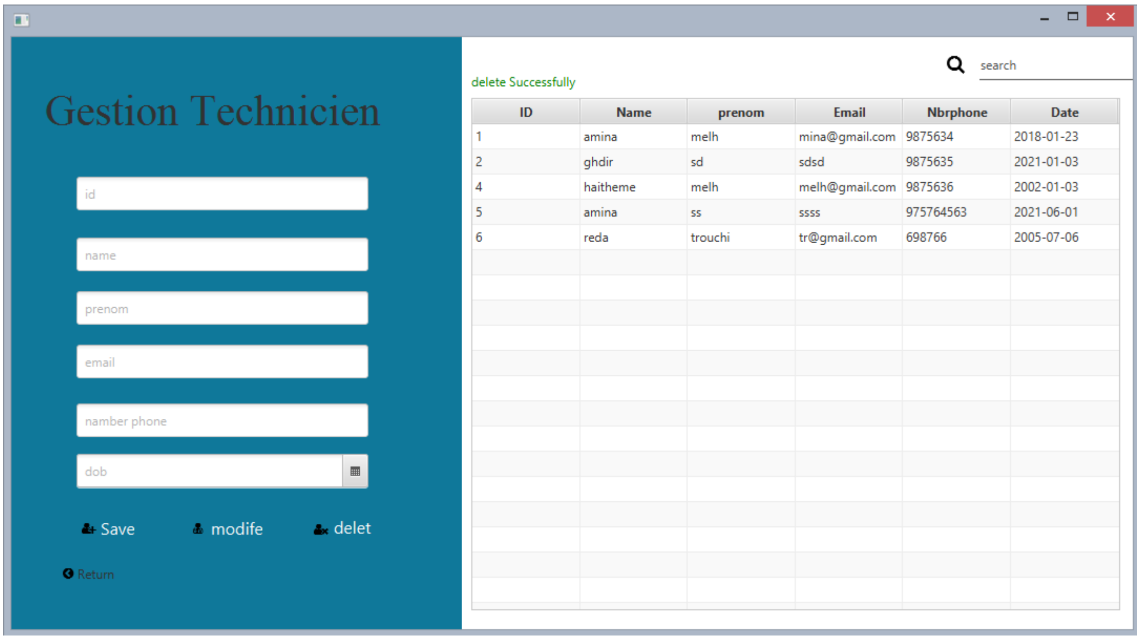
The figure below shows the source code to delete a technician count:



```
@FXML
String DeletData() {

    String st = "DELETE FROM `human` WHERE `human`.`id`='"+ txtid.getText() + "'";

    try {

        preparedStatement = (PreparedStatement) connection.prepareStatement(st);
        preparedStatement.executeUpdate();

        txtid.setText("");
        txtFirstname.setText("");
        txtLastname.setText("");
        txtEmail.setText("");
        txtnumberphone.setText("");
        txtDOB.setValue(null);

        lblStatus.setTextFill(Color.GREEN);
        lblStatus.setText("delete Successfully");
        updateTable();
        return "Success";

    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
        lblStatus.setTextFill(Color.TOMATO);
        lblStatus.setText(ex.getMessage());
        return "Exception";
    }

}
```

Figure 40: *The source code to delete a technician count.*

The figure below (Figure 41) presents the interface to delete technician.

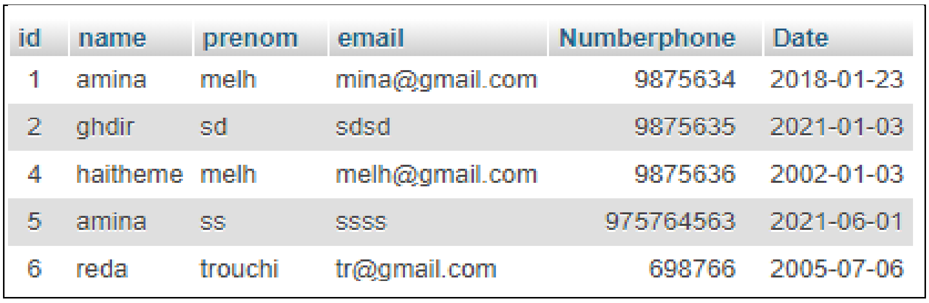


delete Successfully

ID	Name	prenom	Email	Nbrphone	Date
1	amina	melh	mina@gmail.com	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
4	haitheme	melh	melh@gmail.com	9875636	2002-01-03
5	amina	ss	ssss	975764563	2021-06-01
6	reda	trouchi	tr@gmail.com	698766	2005-07-06

Figure 40: *The interface delete technician.*

The figure below (Figure 42) presents the delete of technician in the database.

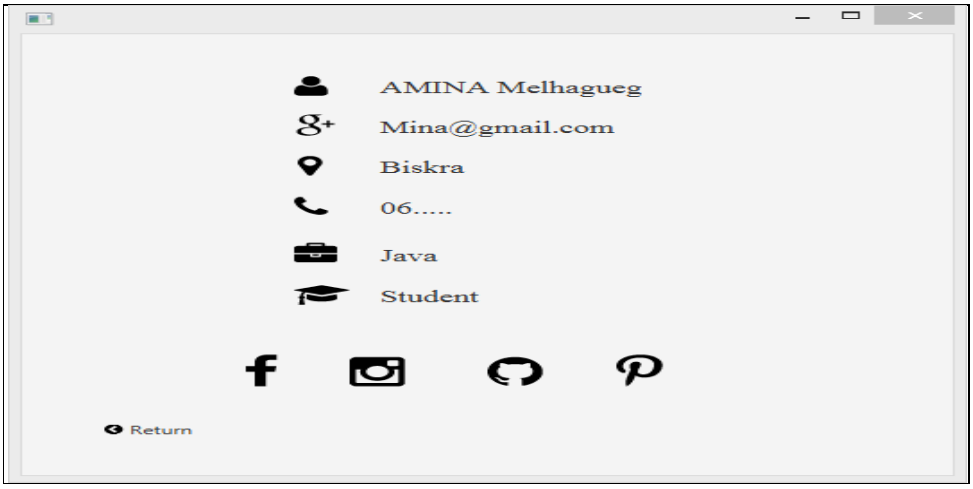


id	name	prenom	email	Numberphone	Date
1	amina	melh	mina@gmail.com	9875634	2018-01-23
2	ghdir	sd	sdsd	9875635	2021-01-03
4	haitheme	melh	melh@gmail.com	9875636	2002-01-03
5	amina	ss	ssss	975764563	2021-06-01
6	reda	trouchi	tr@gmail.com	698766	2005-07-06

Figure 41: *Delete in database of technician count.*

4. The about us interface

The figure below (Figure 42) presents the interface “about us “:



AMINA Melhagueg

Mina@gmail.com

Biskra

06.....

Java

Student

Figure 31: *The interface about us.*