Institute of Information Technology
University of Dhaka

Project name: Assembly Line Management System (ALMS)

Supervised by:
Dr. Ahmedul Kabir
Associate Professor

07/02/2023

Submitted by:
Abhijit Paul (1201)
Mashiat Amin Farin (1202)
Group -2

Submitted to:
SPL 2 program committee

INSTITUTE OF INFORMATION TECHNOLOGY
University of Dhaka

Table of Contents

1. Introduction

This document is a part of our Software Requirement Specification (SRS) for the project "Assembly Line Management System (ALMS)" . In this chapter we will focus on the intended audience for this project.

1.1 Purpose

This document describes the Software Requirement Analysis of our web application Assembly Line Management System(ALMS). It contains all the functional, non-functional, supporting requirements and expected requirements. At the same time it establishes a requirement's baseline for the development of the project. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. But as time goes on, our SRS document is expected to evolve as users and developers work together to validate, clarify and expand its contents. The SRS serves as an official medium of expressing users' requirements to the developer and provides a common reference point for both the developer team and the stakeholder community.

1.2 Intended Audience

This SRS report is intended for several audiences. We discussed with all the possible stakeholders, project managers, developers, designers, and testers; and developed this SRS that gives a clear guideline and documentation to -

A. The users and stakeholders will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
B. The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
C. The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
D. The developers will use this SRS as a basis for developing the system's functionality.
E. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer's documented requirements.
F. The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS.

G. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.4 Conclusion

This analysis of the audience helped us to focus on the users who will be using our software requirements analysis. This overall document will help each and every person related to this project that includes users, project managers, designers, developers, testers, stakeholders to have a better idea about the project

2. Inception of Assembly Line Management System (ALMS)

At the beginning of our project, we enter the inception stage. This stage includes how the project will be started and their **scope** and **limitations**. The main goal of this phase is to **identify the requirements, demand** and establish some sort of **mutual understanding** between the software team and the stakeholders of the ALMS. In order to make this phase effective, we followed some steps namely:

a. Identifying the client of our project
b. Icebreaking
c. Identifying the stakeholders of ALMS
d. Identifying the multiple viewpoints of stakeholders

2.1 Identifying the client of our project

First, we have identified the location from where we will start our expedition. Normally, any hat operates an assembly line could potentially be a stakeholder in our project.
The assembly line industry is broad and encompasses a wide range of sectors, including manufacturing, automotive, aerospace, and more. In these industries, assembly lines are used to produce a variety of products, from consumer goods to complex machinery.

A project to develop a management system for an assembly line could be initiated by the company that owns the assembly line or by an external vendor or contractor. This could include employees, customers, suppliers, investors, and regulators, as well as management and other decision-makers within the company.

In our situation, our stakeholder is Versatile Garments Limited,a ready made garments company located in Ashkona, Dhaka. Their current system of assembly line management depends on a strict chain of command with manual monitoring.

2.2 Icebreaking

Icebreaking refers to diminishing the communication barrier between you and the other person. It is a crucial part since it decides the acceptance of our proposal. We started this phase by talking with them in context-free languages. Their behavior, response to our question or willing to accept the new course management system solely depends on this phase.

2.3 Identifying the stakeholders

Direct Stakeholder

Stakeholders who directly give input to the project or take output from it are direct stakeholders of our ALMS system.

1. Production Manager
2. Line Chief
3. Supervisor
4. Mechanic
5. Quality Assurance
6. Viewer (Company Owner/Higher Management)

Indirect Stakeholder

Stakeholders who indirectly benefit from the system are indirect stakeholders.

1. Ironman
2. Helper
3. Garments Worker
4. Stockholders

2.4 Manual System Story

Current situation of our stakeholder company is shown below:

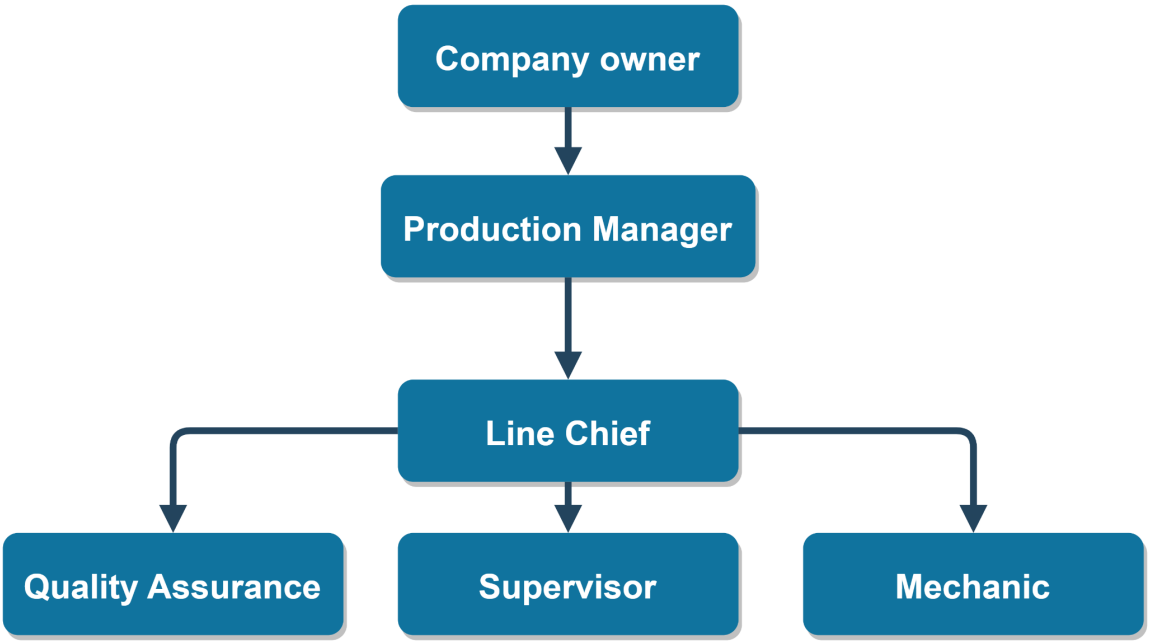


Fig 1: Hierarchy of Assembly Line Management in Versatile Private Ltd.

The company maintains a strict chain of command for communications and assembly line management. However, our system requires only the stakeholders who directly manage the assembly lines, namely:

A. Production Manager
B. Line chief
C. Supervisor
D. Mechanic
E. Quality Checker
F. Company Owner

A. Production Manager

The production manager is responsible for 4-5 assembly lines on the floor. They are responsible for:

1. Allocating Resources to line chief for his assembly line
2. Conveying new product design & planning to line chiefs.
3. Designing layout of assembly lines.
4. Estimating overtime along with the line chiefs.
5. Collecting reports on assembly line
    a. How many pieces were cut. (for cloth)
    b. How many were printed
    c. How much production
    d. Wash sent
    e. Wash received
    f. Finishing receive Count
    g. Post wash sewing count
    h. Last packing
    i. Tagging
6. Handling any issue that occurs in assembly lines.
    a. Handling significant quality assurance issues
    b. Allocating new resources based on availability & necessity.
    c. Ensuring that targets are reached through any means necessary.

B. Line Chief

Each assembly line is assigned to a line chief. All activities in an assembly line are handled by him.

1. Overtime estimation with the production manager in the evening.
2. Congestion control in the assembly line.
3. Handling Mechanic Reports
4. Collecting production reports.
5. Collecting quality assurance reports

6. Handling quality assurance incidents.
7. Monitoring & Handling congestion issues

C. Supervisor

There are two supervisors under line chief. They have the following responsibilities.

1. Counting products at the end workstation of the assembly line.
2. Write down hourly production reports.
3. They mention the machine issues if they notice any.
4. He also mentions any congestion in the assembly he notices to the line chief.
5. Keeping count of **Sampling/Test Products** being sent for washing and retrieving back from washing.
6. Moving workstations to follow the defined assembly line layout.

D. Mechanic

There are two mechanics on each floor. They have the following responsibilities.

1. They are called by supervisors if any machine issues are detected.
2. Machine issues are expected to be handled within 15 minutes.
3. If the machine issue is complex and will require a lot of time to fix, then it is said to the line chief who then requests to the production manager for workstation and 2work reassignment.
4. Note that supervisors notice machine issues and call the mechanic. Machine issues involve:
    a. Sewing wrongly due to misalignment
    b. The machine sews double-stitches wrongly.
    c. The obvious case: needle breaks.
5. Machine checking reports are wasted because these information are not used or forwarded anywhere.

E. Quality Assurance

There are 4-5 quality assurance people on each floor. Each product is checked. Intensive quality assurance activities are done. They also help identify machine issues. Based on order measurement and color designs, they check the product layout, sewing style, shading and cutting measurements for different parts.

1. Quality checking depends on the design of the piece of product. So they often require a reference sample at the beginning phases of quality assurance.
2. This task can be broadly divided into 4 tasks.
    a. Back Part

    b. Front Part
    c. Assembly
    d. Belt Joint
    e. Output
3. If Quality issues exceed tolerance level, then it is reported to the line chief. He explores the reasons.
4. Main issues in Quality drop are usually:
    a. Unattentive worker
    b. Machine issues
5. Quality assurance reports are generated each hour. Every piece of products is checked.
6. **Shading** checks is a real pain for quality assurance people. It is generally done by checking the cloth numbers.

F. Viewer(Company Owner)

The manual reports are converted to digital format at the end of the day as an excel file and goes to management. The management uses that information to generate reports that the company owner views.

G. Viewer(Higher Management)

The higher management collects the excel files and works on them.

2.4.1 Example Workflow For a new Garment Product

First, the cutting department cuts the parts based on that for approximately 100 pieces and sends the pieces to the production department, where the production manager sets up a layout for the assembly line based on the product. And then the workers stitch up that 100 pieces of clothing.

The final test products are measured after sewing. Then the product is washed and measured again. This usually results in a 5-10% error in measurements. That requires the production cutting sizes to be corrected. Then the product is yet again sewn, measured, washed, and measured accordingly. This process is performed by supervisors & quality checkers.

The final measurement is sent to the cutting section for bulk production. And then the measurement and cut cloth bundles are sent to the production unit.

Then the layout of the assembly line is set based on the unit of products that need to be produced by the project manager. On the first day everyone is assigned a specific task and taught that task.

The supervisors aid in the teaching process. Manpower management is performed by the line chief at the beginning of each day.

For the duration of the product being made, the layout is the same. In case of absent or extra workers, the workers are assigned from other assembly lines if needed, or from the same line (if required), or the target is reached during overtime.

Everyday, the machines are checked to ensure that they are working properly. Every worker in the line has hourly production targets, which need to be filled. During each hour, supervisors go to each worker and hear the production report from them. It is then written on paper which is then collected and aggregated by line in charge.
Faulty products are traced back to the worker responsible for it.

Assembly lines are continuously monitored for congestion, faulty machines, quality issues, and identify over-manpower or over-workstation to continuously perform line balencing.
Supervisors, Quality Checkers & Line Chief together perform these monitoring.

Production reports are generated each hour. Production report contains:

- Per hour employee production report
- Done products & Target counts
- Machine problem
- Cutting problem

At the end of the day, If targets are not fulfilled, the production manager makes a meeting with line chiefs and determines how much overtime is needed and any reallocation of resources to fulfill the target.

After all of that, production reports are sent to the IT department to write them down in excel files and then, those files are sent to management who makes meaningful decisions & reports that are sent to the Company Owner later on.

Note that, Congestions are detected manually by the people responsible for the assembly line. If any machine fault or problem occurs, the worker raises their hand and the problem is dealt with, by the supervisor/line chiefs.

The mechanics also perform regular inspections of the workstations.

2.5 Stakeholders' ViewPoint

2.5.1 Viewer(Company Owner's View Point)

1. Long latency of production report reaching the top. It is the drawback of manual reporting. He wished he could see every production report live.
2. Assembly Line Intelligence information on the assembly line.
3. All machine maintenance reports will be used to generate information. This gives insight on the overall workstation condition of the assembly line.

2.5.2 Viewer(Higher Management's View Point)

1. The higher management wants to be able to collect all the information generated by the system so that they can later use it for their work.
2. Prepare reports for morning's planning meeting and evening's overtime estimation meeting on the assembly line.

2.5.3 Production Manager

1. Set production targets for the day.
2. **Define Line Layout:** Total production is set. Different workstations produce different amounts of products per hour. Using all that information that the software will show, the production manager will design the assembly lines. Information about the parts of the cloth design needs to be inserted first into the system. It will also aid, So workstation + number of products + parts information together will aid in defining layout.
3. Define types of different machines and number of those machines available. They are usually:
    a. Sewing machine: Overlocking sewing machine, Electronic Sewing Machines, Single Needle Lock Stitch Machine
    b. Assembly table
    c. Marking table
    d. Shading alignment table
4. Define tolerance of quality drop. (Usually 3~5%)
5. Production manager inserts the **Cloth Design Analysis** file into the system. The file includes: Name of pieces, number of workers needed to make that piece, approximate amount of time to make the piece, workstation needed to make that piece, measurements.
6. **Overtime determination:** At the end of the workday, line chief & production managers sit together to determine how much overtime is needed to complete the rest of the worker. Workstations are assigned to complete the rest of the task. System calculates an approximate **overtime report** to aid in the task.
7. Get requests on workstation resources from line chiefs.

8. Assign tasks to each workstation. Multiple workstations can perform the same task. The tasks are usually of the following types:
    a. Suspender Topisis
    b. Pair Production
    c. Front Rise Overlock: Complex phase. Stacks are positioned in front of the machine.
    d. Form deep & Shoulder joint Estimation
    e. Front pacing Overlock: Very complex
    f. Suspender Join Mark
    g. Suspender Join
    h. Front Pacing Join
    i. Front Shoulder Overlook
    j. Front Placket Tuck
9. He can have 4-5 assembly line lists under his responsibility. Using an admin account, new assembly lines can be added or deleted.
10. His main responsibility as a manager is to listen to people's problems and take appropriate actions. To aid in his task, he will have a log module where he can select the type of event that has occurred and note it down in the system where he writes the daily problems and their solutions he suggested. With time, this document will greatly aid in understanding the assembly line management.
11. Communicate designs with the line chief so that he can decide what training needs to be given.
12. Reports he make:
    a. Target report
    b. Future production
    c. Resource Allocation
    d. Overtime Calculation & Task Assignment

2.5.4 Line Chief

1. Setting production targets for each workstation. (It usually remains the same.)
2. Identify bottleneck areas in the production through constant monitoring and personal experience. Mark those places in the Assembly Line Layout.
3. Automated stack detection to identify congestion.
4. Automatic detection of shading issues.
5. **Dynamic Line Balancing:** The line chief has the information about congestion in the assembly line. So he may decide to change the assembly line layout to accommodate that. And as the day progresses, pressure on the first part of the assembly line decreases and more complex tasks are performed in the last part of the assembly line. So moving more workstations in the last part of the assembly line is necessary. These two tasks are

performed by the line chief in the assembly line layout to handle the bottleneck. Bottleneck theory is used here.

   a. **Stack detection** to identify congestion. The congestion areas will be highlighted in the assembly line layout.
   b. **Bundle detection:** to identify the movement of products.

6. Overtime Determination results and new targets are notified to the line chief.
7. The line chief can change assembly line layout at any time (e.g. to change the layout to accommodate for dealing with design issues. (reverse sewing))
8. Request for new work station if the mechanic sends that it will take more than 15 minutes to fix it.
9. Ensure that zigzag product flows of the assembly lines are working efficiently. If it does not, fix it. This information is inserted into the system which will later be used to generate suggestions for similar assembly line layout.
10. Receive overtime calculation report
11. From the supervisor's report, they estimate whether the line balancing is working properly.
12. Ensure that the target production has been reached through continuous monitoring. These monitoring results come from supervisors mainly. That includes:
    a. Worker reassignment from less-urgent assembly lines to urgent assembly lines.
    b. Monitoring machines & Immediately responding to machine failure.
    c. Continuous quality checking & back propagation.
    d. Checking for bottleneck & overworker points.
    e. Hourly production monitoring
13. **Testing procedure:** Hand-cut design → Line chief → Stitch 10-15 sample → Measure 3 sample → Wash → Iron → Measure Sample → **Pattern Correction based on error** →Stitch 100 sample → Measure sample → Wash → Iron → Measure Sample **Pattern Correction till error margin very low and product approved**
14. Reports he make:
    a. Sample testing report.
    b. Assembly line performance report for the day
    c. Assembly line performance report for the month
15. Reports:
    a. How many pieces were cut. (for cloth)
    b. How many were printed
    c. How much production
    d. Wash sent
    e. Wash received
    f. Finishing receive Count
    g. Post wash sewing count
    h. Last packing

i. Tagging

2.5.5 Supervisor

1. Count total products produced at the last workstation of the assembly line. This is done once per hour. This information is sent to the **Production Report module.**
2. At the end of the assembly line, inspect each product for **shading** issues.
3. Compare with sample and quality assurance
4. Bulk calculation when sending samples for washing.
5. Identify Over-workstation: They monitor the line manually to see any point where there are more workstations than needed. They mark those workstations in the assembly line and this information is sent to the line chief as notification.
6. Identify Under-workstation: Identify location in the assembly line where more workstations are needed manually. Mark those positions in the assembly line layout. This information is sent to the line chief as notification and is used for dynamic line balancing.
7. Let the line chief know if the mechanic does not arrive at a broken machine within 10 minutes.
8. Being able to teach the workers about their work easily in the first day of the assembly line for a new product.
9. Reports he make:
    a. Production Report

2.5.6 Quality Checkers

1. He inspects the quality of products in each workstation. He will write down the result after inspecting the workstation.
2. He also selects the reason for the quality drop for that workstation. Main reasons of bad quality are usually:
    a. Not understanding the job well
    b. Not focusing on the task
    c. Machine issues
3. If a drop of quality exceeds a threshold, he marks the workstation in the assembly line layout. A notification is sent to the line chief on the quality drop. He then physically inspects the workstation for issues or takes what action he sees fit.
4. They often look at the sample product measurements to ensure quality at the beginning of inspecting a new product. They can view this information in the system in the **Product Design Analysis module.**
5. Send faulty products back to the appropriate workstation. Select the workstation in the assembly line layout and the line chief will get a notification of that.
6. If there are significant issues with Q/A, the line chief is notified about this issue of the line.
7. This task can be broadly divided into 4 tasks.

    a. Back Part
    b. Front Part
    c. Assembly
    d. Belt Joint
    e. Output
8. After they are done inspecting a line for the hour, their report is sent to the line chief.
9. Reports he make:
    a. Quality Assurance Report

2.5.7 Mechanic

1. Set a machine “Green” after inspecting it. Green machines will be part of the available resource for project managers to allocate to assembly lines.
2. Each part of the machine is checked. The mechanic selects the machine issue and the approximate time to fix it. If he thinks that it will take more than 15 minutes to fix it, he marks the workstation and the line chief knows about it. He may then request for a new workstation to the production manager.
3. After fixing a machine, he needs to write about its issues. He may select from usual issues like:
    a. Sewing wrongly
    b. The machine sews double-sew wrongly.
    c. The obvious case: needle broke.
4. Insert the model number, day of purchase information about each machine.
5. Insert the issue with the machine and the fix you provided.
6. Reports he make:
    a. Mechanic Report

2.6 Working Towards Collaboration

1. Automatic detection of shading issues: We can not do it properly. It is error-prone. But we will help in identifying the issue by showing a similarity matrix.
2. Count total products produced at the last workstation of the assembly line. This is done once per hour: Perhaps we can automate it. We can set the motion. Move the complete product from right to left. We can count this way.

2.7 Proposed System

From our understanding, we propose the following system to the stakeholders for verification.
https://app.diagrams.net/#G1rWuFNTSb6iKOLFy6JFLeJW7YL7LqR79c

2.7.1 Production Manager

Register Resource: Resources mean Workstations. Workstation can be an assembly table, Overlocking sewing machine, Electronic Sewing Machines, Single Needle Lock Stitch Machine, marking table, shading alignment table.
The following information is needed to register a new resource - Model of each resource, time of purchase, cost of purchase, ownership information, machine type.
Register Assembly Line: The following information is needed to register an assembly line - name of assembly line, number of assembly line etc.

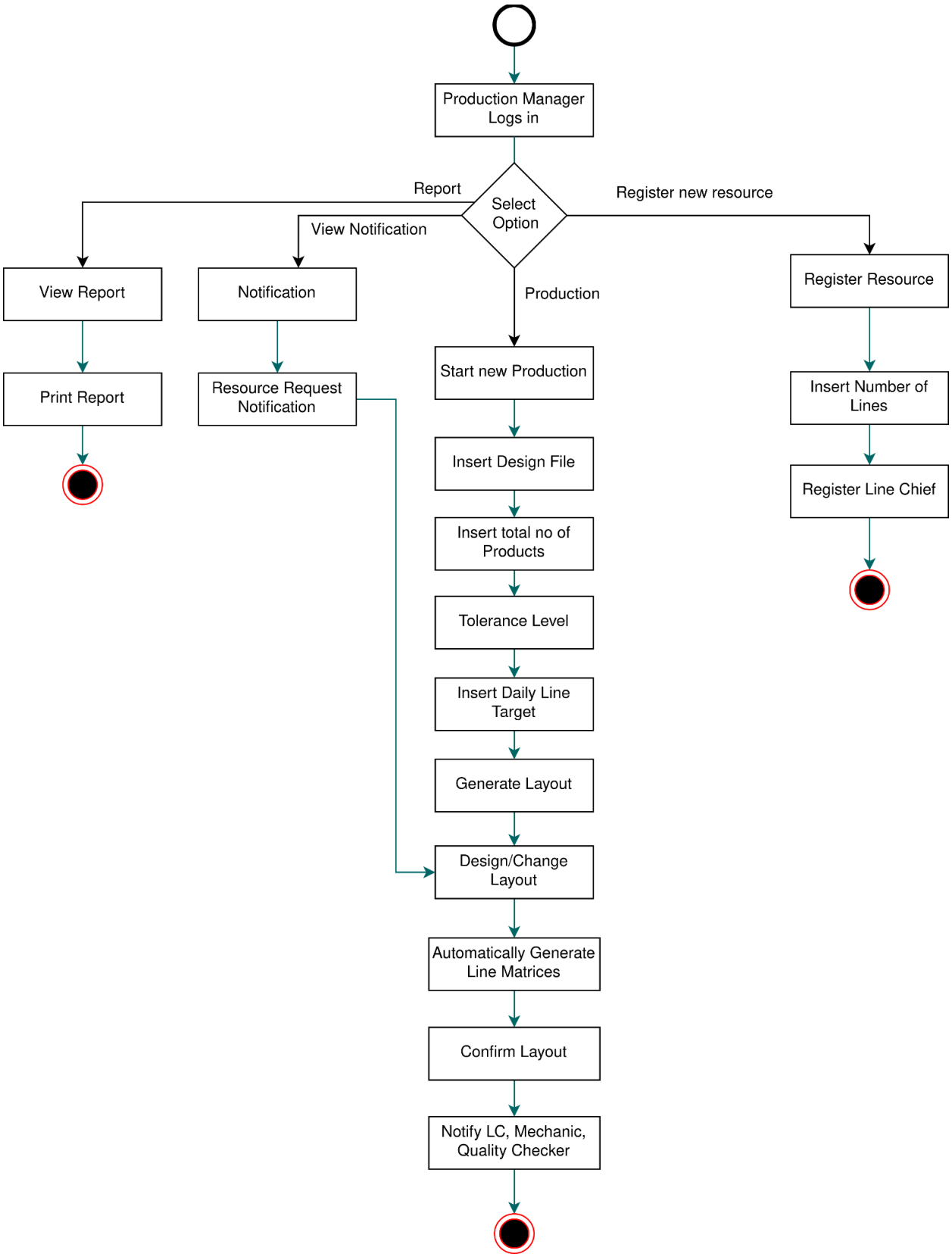


Fig 2: Production Manager - Proposed System

2.7.2 Line Chief

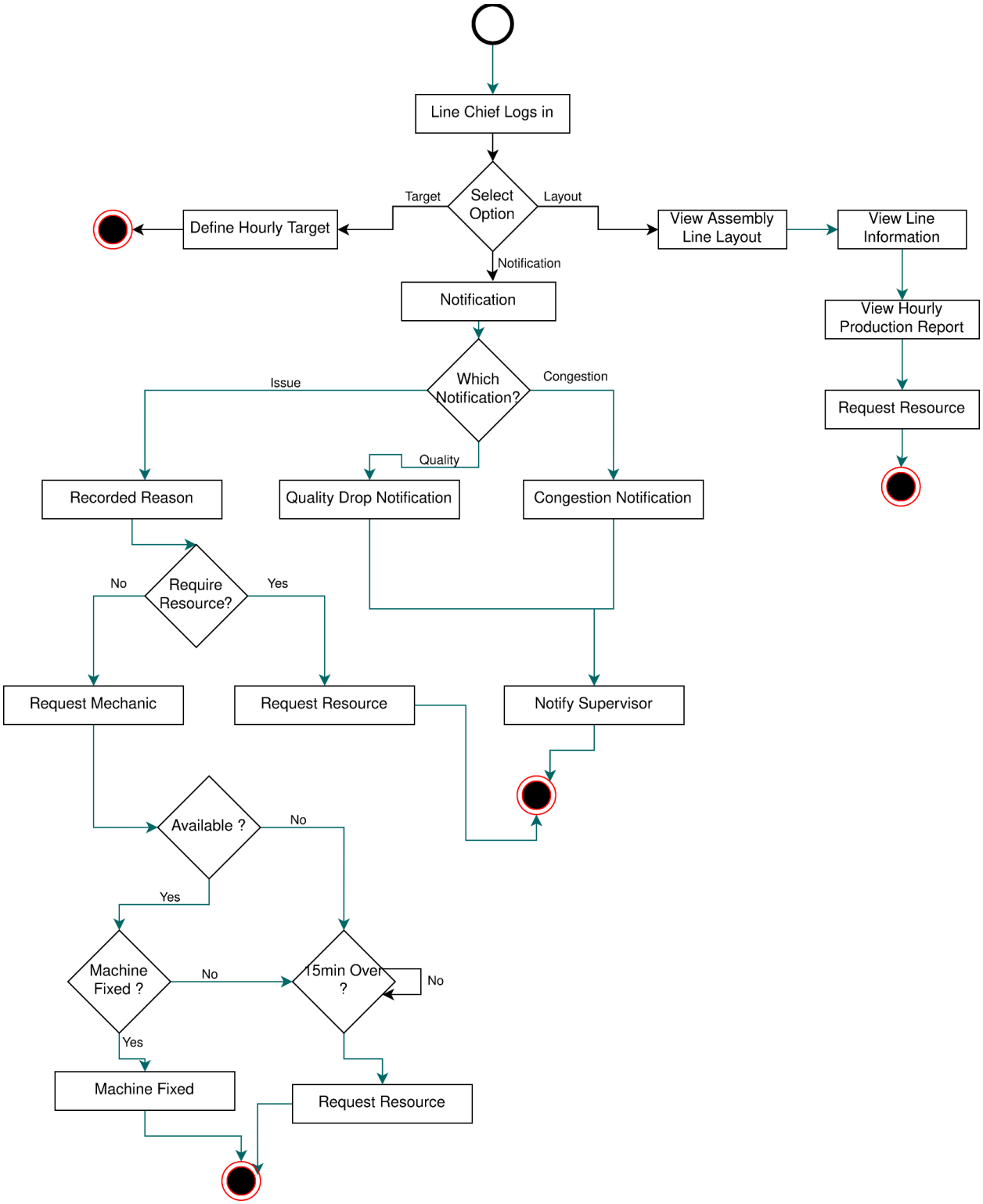


Fig 3: Line Chief - Proposed System

2.7.3 Supervisor

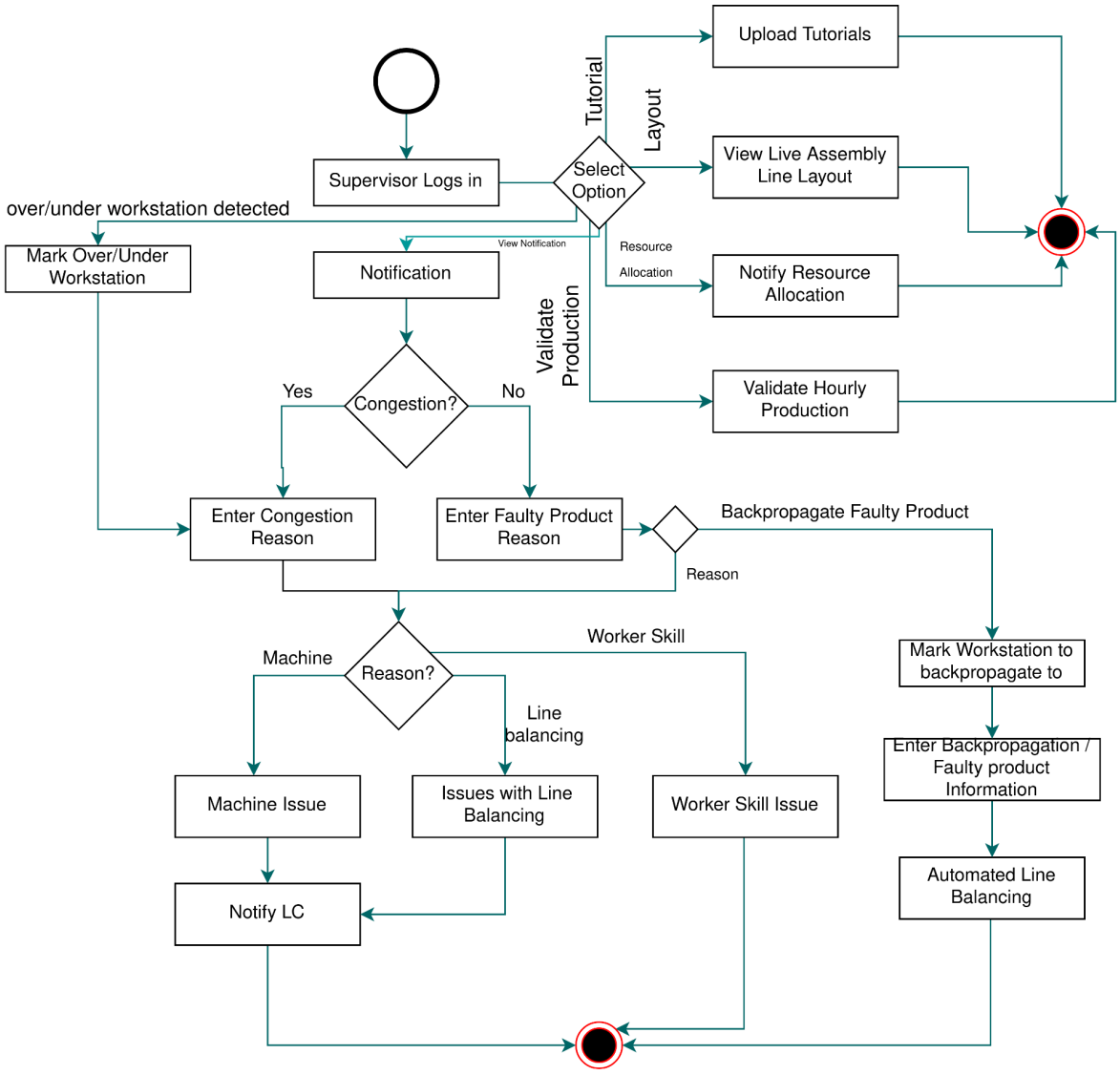


Fig 4: Supervisor - Proposed System

2.7.4 Mechanic

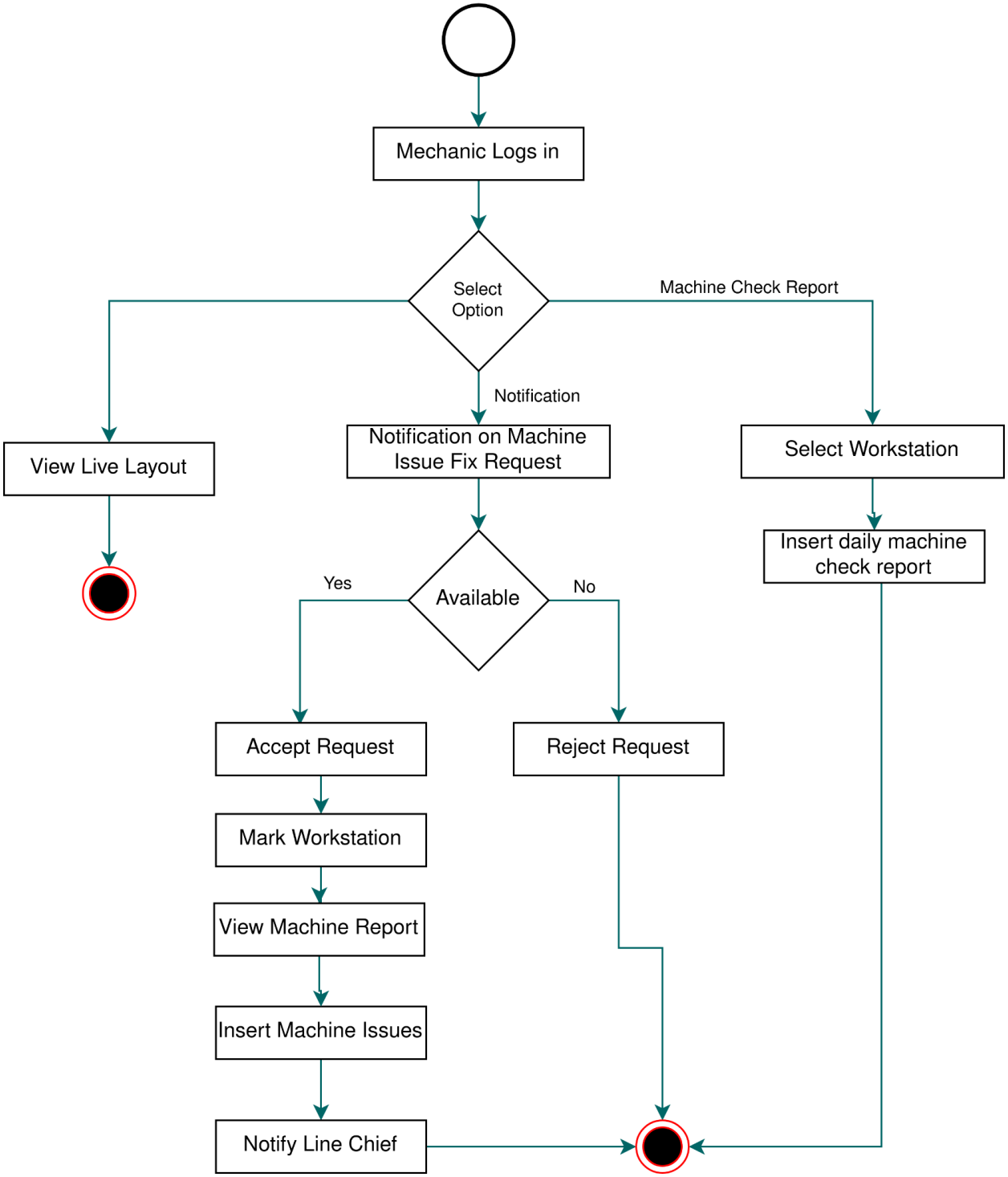


Fig 5: Mechanic - Proposed System

2.7.5 Quality checker

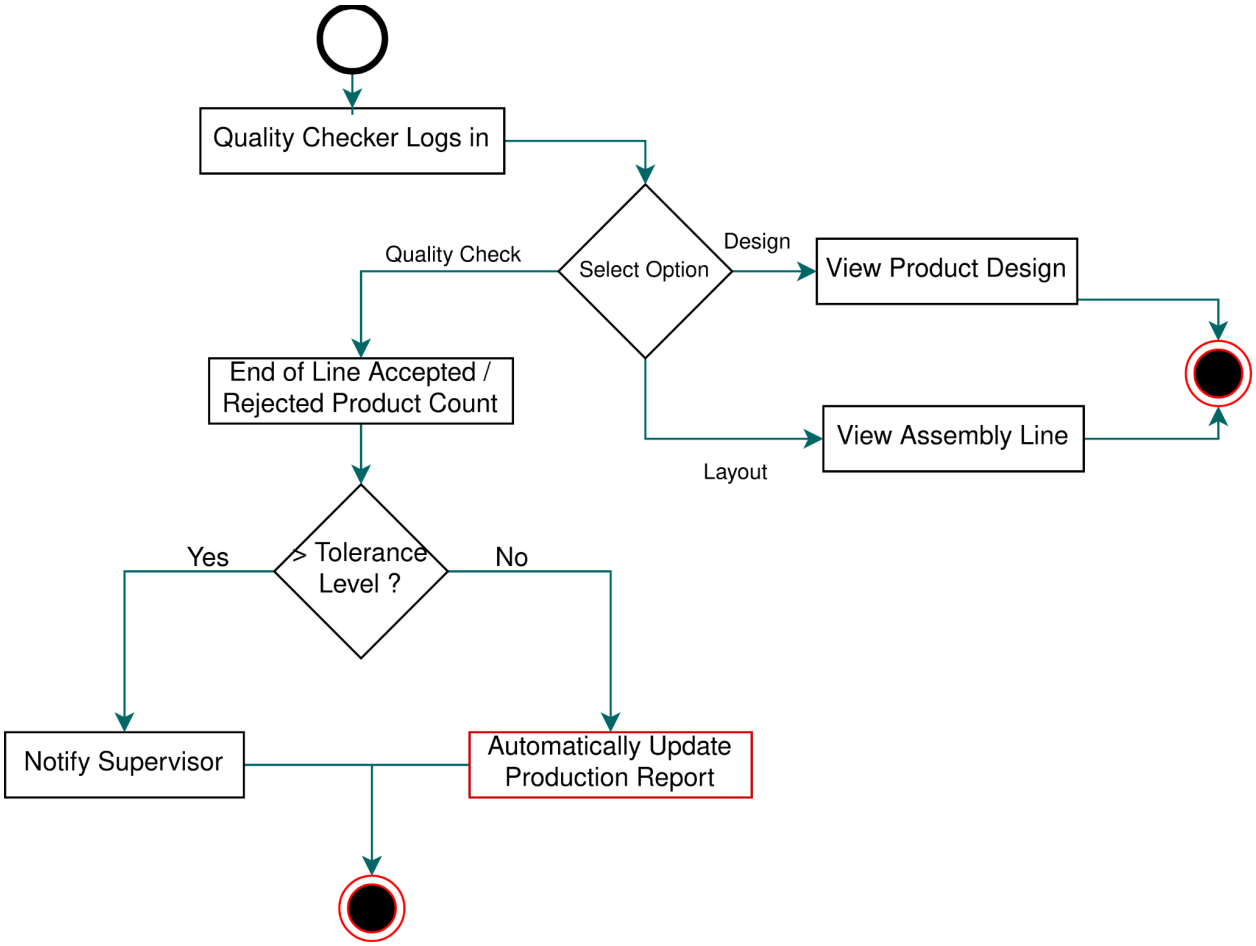


Fig 6: Quality Checker - Proposed System

2.7.6 Viewer

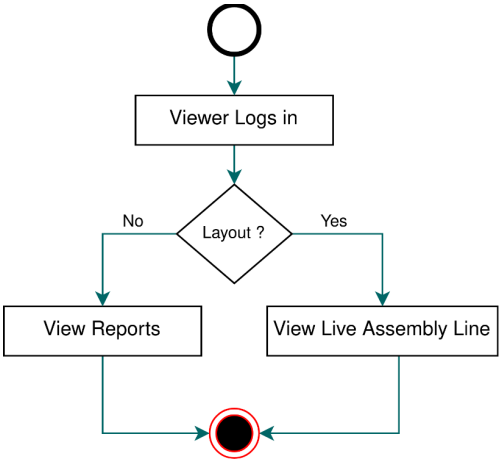


Fig 7: Viewer - Proposed System

2.7.7 Guest user

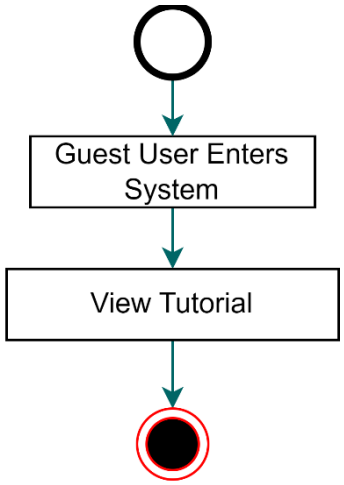


Fig 8: Guest user- Proposed System

2.8 Conclusion

Our primary goal is to design a software which will make assembly line management easier and more efficient, helping both management and workers to fulfill their potential and achieve higher productivity, helping the entire company environment. At the same time, a platform where information exchange between all parties can be streamlined and managed efficiently. Any congestion in the assembly line can be detected automatically and increasing efficiency of the assembly line will also increase productivity.

The software to be built must be designed in such a way that it will not be complicated for users and will be simple enough so that users are able to maintain it without annoyance. Otherwise there will be a risk of users not adopting the system.

3. Elicitation of ALMS

After discussing the Inception phase, we need to focus on the Elicitation phase. So this chapter specifies the Elicitation phase.

3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the stakeholders. We have faced many difficulties, like understanding the problems, making questions for the stakeholders, limited communication with stakeholders due to shortage of time and volatility of the stakeholders. Though it is not easy to gather requirements within a very short time, we have surpassed these problems in an organized and systematic manner.

3.2 Eliciting Requirements

We have seen in the previous chapter, where the inception phase of requirement engineering has been described. The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. We have finished the following tasks for eliciting requirements-

- Collaborative Requirements Gathering
- Story
- Quality Function Deployment

3.2.1 Collaborative Requirements Gathering

We have met with our stakeholders and clients in the inception phase. The meetings created the requirements for this application. Initially the meetings created an indecisive state, from which, over more elaboration, we came up with the solution which is acceptable to both the client and stakeholders.

3.2.2 Story

The admin user is the default user of the system. The admin user adds users(production manager, line chief, mechanic, supervisor, quality checker, viewer etc) to the system by setting their username, title and general description. The admin sets a default password for the created user. The admin can edit their information anytime. The users can later change their password.

The production manager logs in to the system after authentication. He then inserts total resources (Overlock sewing machine, assembly table, marking table, shading alignment table, electronic sewing machine) available to him. He then inserts the number of assembly lines under his control and their general information (line chief responsible for it, types of machine in the line). He then inserts the design file for a new product. The design file contains the types of tasks to be done on the product, the types of machines required to do the tasks, the number of machines required for the tasks, the sizes of the cutting pieces and final product. He then inserts the number of products to be produced. He then selects an assembly line, sets the number of products he wishes to produce in that line and defines the assembly line layout using an interactive layout interface. The system first generates an assembly line layout based on available information. The production manager can change the layout if he wants. The system shows useful matrices(e.g. Line efficiency, Resource Utilization Index, Possible Bottleneck points) based on his tweaked layout. After that, the production manager confirms the layout. System sends a notification to the line chief of the respective assembly line about the new layout and the targets set for the assembly line. Production manager also defines a tolerance level of error for the product.

After logging in, the production manager can assign a line chief to one assembly line.

Line chief logs into the system. He views the assembly line layout. The line chief then defines hourly targets for each workstation of the assembly line layout. After that, a notification is sent to supervisors & mechanics on the approved assembly line layouts.

After that, production of the day starts. The assembly line layout includes workstations, the piece of product to be produced in that workstation & target for one hour. The supervisors can upload tutorials for each workstation's works that the workers may see as a **guest** user of the system.

The assembly line is functioning now. So the supervisors, line chief constantly monitor the line for any issues. The system aids in the process by monitoring the assembly line stacks using a visual sensor. If the stack height reaches a certain limit, then the workstation is identified as congestion and the assembly line layout is updated to mark the congested workstation and a notification is sent to line chief. Supervisor can also mark a workstation congested.

(Workflow A start) Line chief then instructs the supervisors to go & check the reason for congestion through the system. In this case, the supervisor gets notification, checks the workstation & records the reason for congestion. The reasons are usually

1. Worker's learning issues of the new task.
2. Machine Issues: For which the line chief is notified, who requests a mechanic to fix it. The mechanic is instantly notified. The mechanic may accept or reject the request. If accepted, the mechanic can view the assembly line layout and where the problem has occurred. After fixing the machine, he inserts the failure reason. If a mechanic is not available within 15 minutes, then a notification is sent to the line chief. The line chief requests for a new workstation (in place of the broken one) and the production manager gets the request as notification. He either accepts or rejects it based on available resources.

3. Line Balancing Problems: For which supervisor requests for more workstations to the line chief. The line chief gets the notification. He can validate the request and forward the request to the production manager. The production manager gets the notification, reviews it and accepts or rejects the request Additionally, the system also generates line balancing suggestions considering backpropagation information. (workflow A end)

If over-workstation issues are detected, supervisors mark the workstation and a notification is sent to the line chief. The line chief then performs appropriate line balancing in the assembly line layout and requests more resources if necessary.

Mechanics check the workstation daily and insert whether they have any issues. This and the workstation fixing information generates the machine reports.

At the end of the assembly line, quality checkers will check all the products and input the number of accepted and rejected products. From this the hourly production report will be generated, which the supervisor will validate. If there is any rejected product - the workstation responsible for it is marked by a quality checker and the product is **back-propagated** to fix the issue. This backpropagation information is stored in the system and can be leveraged for dynamic line balancing, as the load on each workstation now can increase uncertainty.
Based on the hourly production report - if the quality exceeds the tolerance level set by the production manager, then an alert notification is generated. In this case, the LC is notified and he notifies the supervisor to inspect and insert the issue of the quality drop from a few options. Then *workflow A* is in motion once again. If it's some other issue, the line chief handles them manually.

After being done with counting, the hourly production report is instantly sent to the line chief. The line chief then reviews the overall production and requests resources in assembly line layout

if necessary. In such a case, a notification is sent to the production manager for his approval. Additionally, the supervisor gets notification of new allocation of resources.

At the end of the workday, the daily production report is sent to the production manager. Production report includes total number of productions vs target, issues that happened throughout the day, quality index, machine issues index and required overtime based on the number of production deficit and hourly production rate. Note that overtime is estimated considering the target of the assembly line and products produced so far.

The live layout of the assembly line is broadcasted in the system for any viewer. The assembly line layout includes: target per workstation, quality issues per workstation, assembly line performance matrices, target vs achieved production, live congestion results and tutorials for workers.

The following reports are generated and sent to the production manager daily. They are also streamed in the system for all viewer users. Note that the admin can set any information he wish the viewers to see.
- Hourly Production Report: Hourly completed product, hourly rejected product. Start time and end time of each stack and average cycle time.
- Daily Production Report: Production per hour of the day, Target Production, Reached Production, Reasons behind fall of production, allocated workstation, assembly line layout. Target Vs Reached production, system suggested overtime. Note that, reasons for fall of production are collected from overall congestion data, quality issues data.
- **Quality Assurance Report:** Tolerance level, causes of quality fall, Assembly line graph of quality drop. This graph is generated using quality drop estimation of each hour. Note that the system gets quality issues before generating it.
- **Machine Report:** How many machines broke, how many machines fixed, time wasted, machine status.
- Congestion Report: Congestion graph of the assembly line layout, congestion count, efficiency vs congestion graph.
- **Line Balancing Report:** Bottleneck points graph, time series graph.

Note that, the following reports are not sent daily:
- Monthly Assembly line performance report

The production manager may print the reports. The information is archived after 4 months.

Metrices
Line Efficiency Index = number of products produced / Amount of time needed

Utilization Percentage = Actual Number of Hours Worked (by a particular resource) divided by the Total Available Hour to use the resource.

3.2.3 Quality Function Deployment

After talking & negotiating with the stakeholders, we have finalized the following requirements for our system. They are:

Normal requirements

1. The admin can add users to the system.
2. The user can change their user information.
3. Admin can add viewer information.
4. The users can change their account password.
5. Production manager, line chief, supervisor, mechanic, quality checker, admin, viewer and guest users have their own login system.
6. Production managers can input the amount of resources that is available to him.
7. Production manager can upload product type and the different types of tasks required to make the product as the design file
8. Production manager can create assembly line layout
9. Production manager can assign the line to a line chief
10. Production manager defines: the number of products to be produced, daily line targets, fault tolerance levels.
11. Production manager finalizes the assembly line layout.
12. Production manager is notified on hourly production report, request for resource, congestion information
13. Line chief can view layout and line targets
14. Line chief can set hourly line target
15. Line chief can request for extra resources.
16. Line chief will be notified on quality issues, hourly production report, congestion information
17. Line chief can request mechanics to fix workstations.
18. Supervisors can view assembly line layouts and product details.
19. Quality checker inserts production count, and supervisor validates the count
20. Supervisor uploads congestion reason
21. Supervisor uploads backpropagation/faulty product information
22. Supervisor can mark over/under workstations of the line.
23. Supervisors upload tutorials for workstations.
24. Mechanics accept/reject workstation fixing requests and insert workstation information(fixed or not, what type of problem occurred)

25. Mechanics uploads daily workstation details
26. Quality checkers upload reasons for quality drop.
27. Viewers can view live assembly line status, production reports and other informations made available by the admin
28. Guest users can view tutorials.
29. The production manager can download reports.

Expected Requirements

1. Each user will have a different level of access to the system.
2. Notification system to notify about any new reports or messages from other users.
3. Production manager can generate performance matrices on assembly line layout
4. The system must have an admin account.
5. Reports will be digitally generated automatically.
6. Live layout of an assembly line must be updated and viewable at all times
7. Communicate all the reports (production, overtime, washing, sample testing, quality assurance, machine report, congestion report, line balancing report, daily assembly line performance report) to the concerned users

Exciting Requirements

1. Automated Line balancing based on product
2. Congestion detection in the line based on object/stack detection.
3. Generate layout based on design file and resources

3.2.4 Usage Scenario

ALMS is an assembly line management system. The application will help the users streamline their processes and make the assembly line more efficient and productive.
The application will have the following features:

User management

There will be users of the following type:

a. Admin account
b. Production manager
c. Line Chief
d. Supervisor
e. Mechanic
f. Quality assurance
g. Viewer (Company owner and higher management)
h. Guest users (employees)

The stakeholders will hold accounts, which can be used to denote their functionality and authorization level.
There will be the following types of user accounts:

Admin Account
An administrator will act as the background employee to maintain the whole system. The managing director is also an admin. Admin can add all users to the system.

Production manager account: Production manager accounts will be created by the administrator. The following information will be required while creating the account:

A. Name
B. user ID
C. Email
D. Password
E. Assigned Floor

Line Chief account: Supervisor accounts will be created by the administrator. The following information will be required while creating the account:

A. Name
B. user ID
C. Email
D. Password
E. Assigned Floor

Supervisor Account: Supervisor accounts will also be created by the admin. It requires the following information:

a. Name
b. user ID
c. Email
d. Password
e. Assigned Floor

Mechanic Account: Mechanic accounts will also be created by the admin. It requires the following information:

a. Name
b. user ID
c. Email
d. Password
e. Assigned Floor

Quality Assurance Account: Quality assurance accounts will also be created by the admin. It requires the following information:

a. Name
b. user ID
c. Email
d. Password
e. Assigned Floor
f. Assigned line

Viewer: Viewer accounts will also be created by the admin. It requires the following information:

a. Name
b. user ID
c. Email
d. Password
e. Designation

Guest user: Guest user accounts will also be created by the admin. It requires the following information:

a. user ID
b. Name
c. Email

The login system will be separated into sign up, login and guest user. All users must create an account to access the system, except from guest users.

Sign up: To access all features, the user must create an account.
Log-in: The user can log-in using his email/username and password.
Change account information: Users can change account information and their passwords.

Layout Management

Floor information: Production manager inserts total resources (Overlock sewing machine, assembly table, marking table, shading alignment table, electronic sewing machine) available to him. He then inserts the number of assembly lines under his control and their general information (line chief responsible for it, types of machine in the line).

Layout Generation: When there's an order for a new product, the production manager inserts the design file for a new product and the number of products to be produced. The design file contains the types of tasks to be done on the product, the types of machines required to do the tasks, the number of machines required for the tasks, the sizes of the cutting pieces and final product.

He selects an assembly line, sets daily production targets, the tolerance level of error and defines the layout of the assembly line using an interactive layout interface. The system first generates an assembly line layout based on available information (from design file). The project manager can make changes in the layout if he wants. The system shows useful matrices(e.g. Line efficiency, Resource Utilization Index, Possible Bottleneck points) based on his changed layout. After that, the project manager confirms the layout design. System sends a notification to the line chief of the respective assembly line about the new layout and the targets set for the assembly line.

The line chief can view the assembly line layout. The line chief then defines hourly targets for each workstation of the assembly line layout. After that, a notification is sent to supervisors & mechanics on the approved assembly line layouts.

Resource allocation: Extra resource allocation requests sent by the line chief are validated by the production manager, after which the assembly line is edited and line chief, supervisor and quality checker is notified. Production manager can also change the layout based on production reports.

View line and product: Production manager, line chiefs, supervisors and quality checkers can view the assembly line layout and the product design and measurements at all times. The live layout of the assembly line is broadcasted in the system for any viewer. The assembly line layout includes: Production per workstation, target per workstation, quality issues per workstation, assembly line performance matrices, target vs what production has been achieved, live congestion results and tutorials for employees.

Congestion control

Problem detection:

The system monitors the assembly line stacks using a visual sensor. If the stack height reaches a certain limit, then the workstation is identified as congestion and the assembly line layout is updated to mark the congested workstation and a notification is sent to line chief. Supervisor can also mark a workstation congested. Based on the quality assurance report, workstations can be marked as congestion as well.
In all cases the line chief is notified. Line chief notified supervisor and supervisor insets the reason for congestion/quality error:

1. Worker learning issue
2. Machine issue
3. Line balancing problem

The LC is notified of the reason.
To solve the problem, the following actions are in place:

Teach employee: Supervisors upload tutorials of tasks on the system, which workers can view as guest users of the system.

Call mechanic: For which the LC requests a mechanic to fix it. The mechanic is instantly notified. The mechanic may accept or reject the request. If accepted, the mechanic can view the assembly line layout and where the problem has occurred. After fixing the machine, he inserts the failure reason. If a mechanic is not available within 15 minutes, then a notification is sent to the line chief. The line chief requests for a new workstation (in place of the broken one) and the production manager gets the request as notification. He either accepts or rejects it based on available resources.

Line balancing problems: For which line chief requests for more workstations to the production manager. The production manager gets the notification, reviews it and accepts or rejects the request.

Supervisor can also detect underworked stations and overworked stations and mark them to notify the LC.

Quality reports: At the end of the assembly line, quality checkers will check all the products and input the number of accepted and rejected products.If there is any rejected product - the workstation responsible for it is marked by a quality checker and the product is **back-propagated** to fix the issue. This backpropagation information is stored in the system and can be leveraged for dynamic line balancing.
Based on the hourly production report - if the quality exceeds the tolerance level set by the production manager, then an alert notification is generated. In this case, the LC is notified and he notifies the supervisor to inspect and insert the issue of the quality drop from a few options.
Mechanic reports: Mechanics check the workstation daily and insert whether they have any issues. This and the workstation fixing information generates the machine reports.

Report Generation

Hourly production report: At the end of each hour, the supervisors validate the count of products made by the quality checkers. After being done with counting, the hourly production report is instantly sent to the line chief. It will include the number of products produced in an hour, validated by the supervisor. It will also include faulty workstation incidents, workstation responsible for rejected products. The hourly report will be viewable by all users.

Daily production report: At the end of the workday, the daily production report is generated, which includes: total number of productions vs target, issues that happened throughout the day, quality index, machine issues index and required overtime based on the number of production deficit and hourly production rate.

Machine report: How many machines broke, how many machines were fixed, daily status of workstation will be included in this report. Mechanics will check the workstations before the start of the day and input whether the machines are satisfactory or not. If the machine is not satisfactory - he will note the reason and notify the line chief.
During the day, the mechanic checking the broken workstation will note down the reason and fix it. If it is not fixable, he will notify the line chief.

Quality assurance report: will include the accept/reject ratio of products, the workstations responsible for rejected products. Workstation accept/reject ratios, causes of quality fall, Assembly line graph of quality drop

Congestion Report:This report will contain congestion graph of the assembly line layout, congestion count, efficiency vs congestion graph

Line balancing report: This report will contain Bottleneck points graph, time series graph

Monthly report: summary of the daily reports

Notification
All types of notifications to production manager, line chief, supervisor, quality checker and mechanic will be made through this.

4. Scenario based modeling of Assembly Line Management System (ALMS)

This chapter describes the Scenario Based Models for Assembly Line Management System (ALMS)

4.1 Introduction

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

4.2 Use case diagrams

Use case defines the stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators.

Primary Actors

The primary actors of our system are:

a. Production manager
b. Line chief
c. Supervisor
d. Quality checker
e. Mechanic
f. Admin

Secondary Actor

The secondary actors of the system are the following:

a. Viewer (Company owner and higher management)
b. Guest users (employees)
c. Video sensor

Use case diagram link:
https://drive.google.com/file/d/15-U4UD-OwocEfyJJgMupuZXnqWerQ2p2/view?usp=sharing

Level 0

Use case name: Assembly Line Management System
Primary actor: Production manager, Line chief, Supervisor, Mechanic, Admin, Quality checker
Secondary actor: Viewer, Guest users, Video sensor

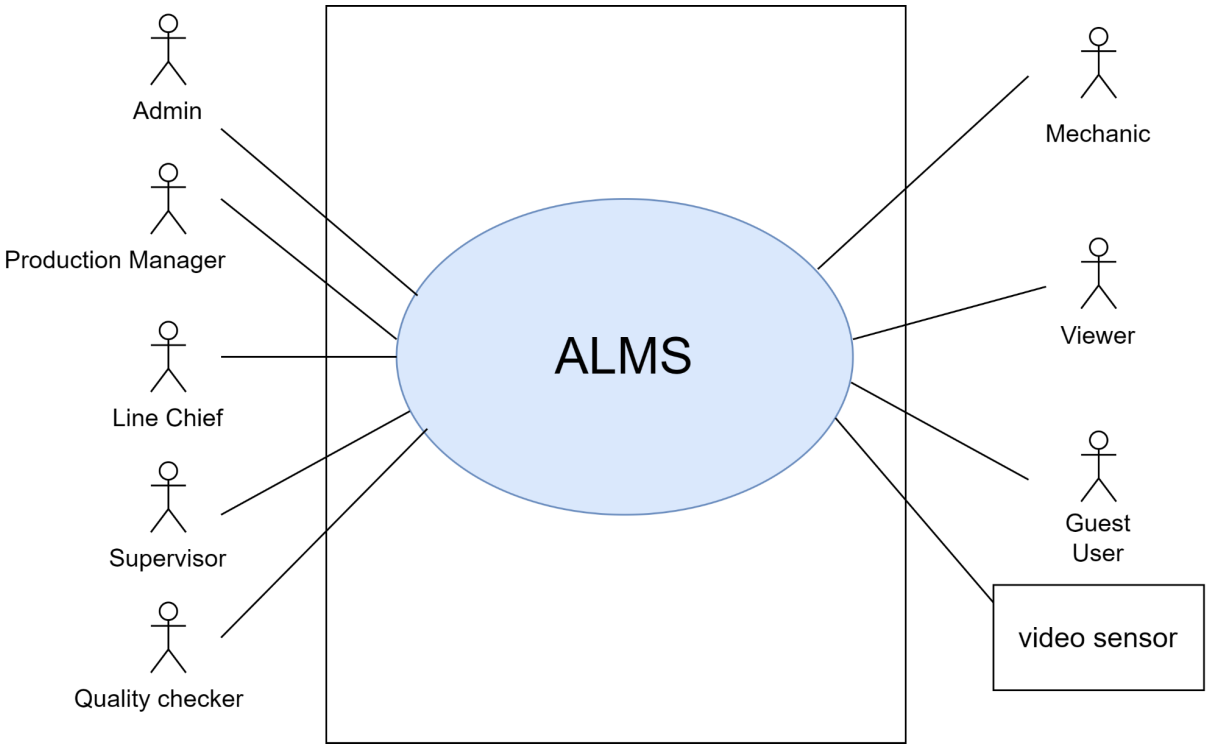


Figure 9: Use case diagram level 0: Assembly line management system (ALMS)

Description: This use case shows the low-level interaction between system and actors.

Level 1

Use case name: Assembly Line Management System
Primary actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic, Admin
Secondary actor: Viewer, Guest users, Video sensor

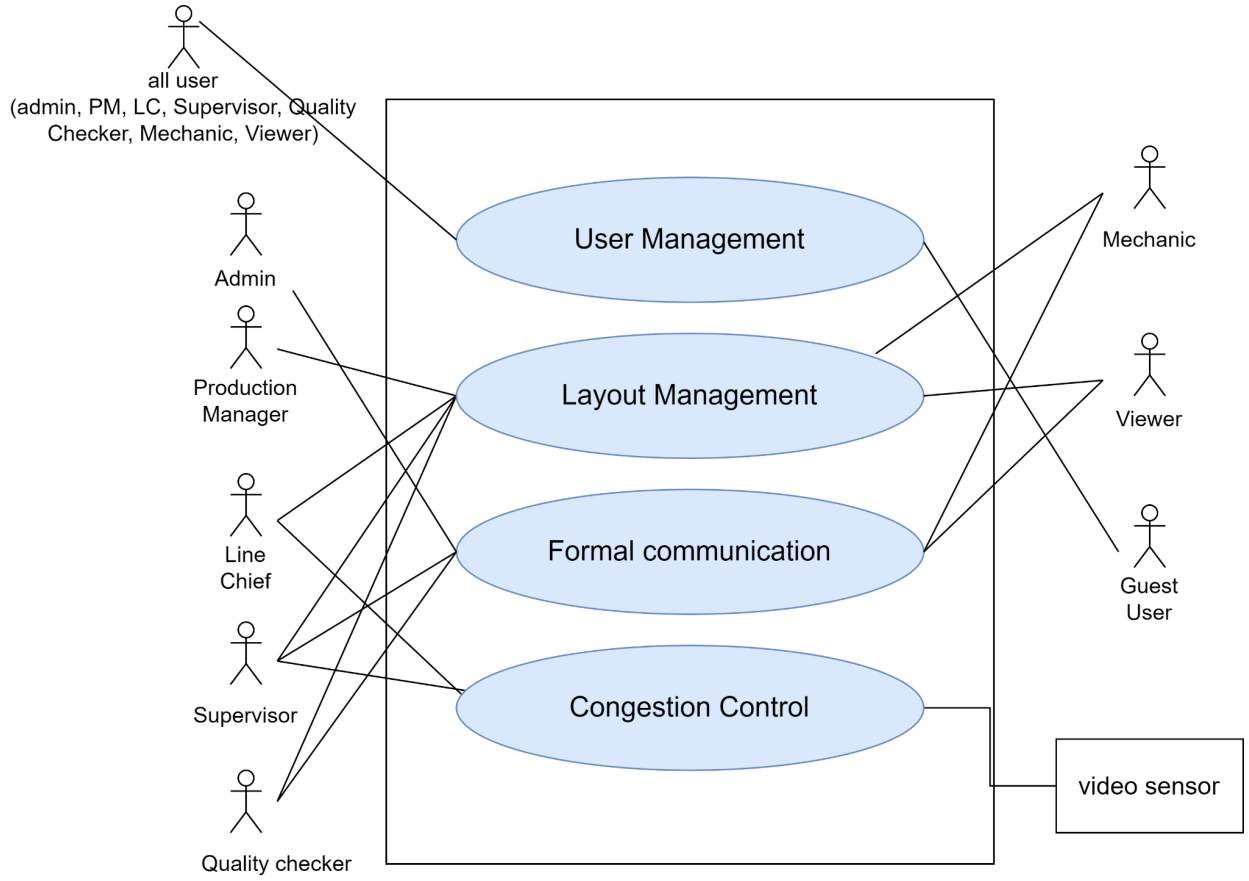


Figure 10: Use case diagram level 1: Assembly line management system (ALMS) details

Description: From this level all the subsystems of the proposed main system and connectivity of those subsystems through actors has been explicitly explained. From this level interaction between actors and subsystems will be clearer. Here, the whole system is divided into four subsystems.

Level 1.1

Use case name: User management
Primary actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic, Admin, viewer, Guest users

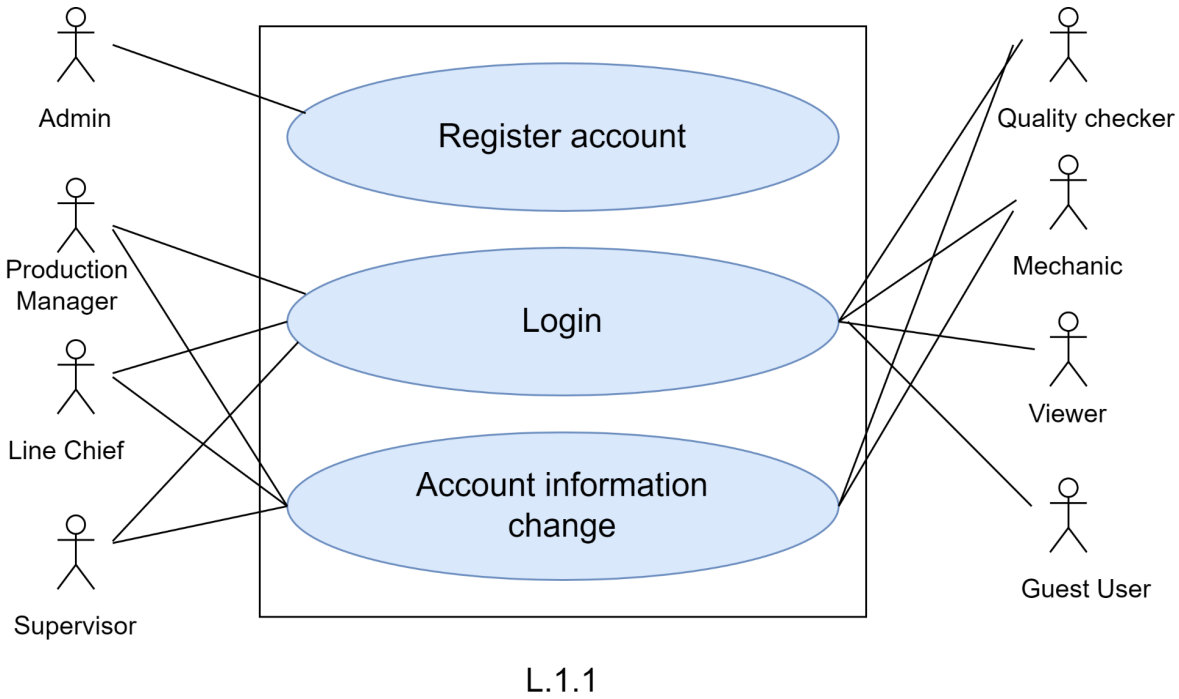


Figure 11: Use case diagram level 1.1: User management

Description: The whole login and registration process has been explicitly explained in this level in three more sub systems. They are registration, log in & recover accounts.

Level 1.1.1

Use case Name: Register
Primary Actor: Admin
Secondary actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic, viewer
Description:
Action Reply:
A1: admin registers account for users
R1: users can now log in
A2: admin changes user information based on request
R2: information is updated

Level 1.1.2

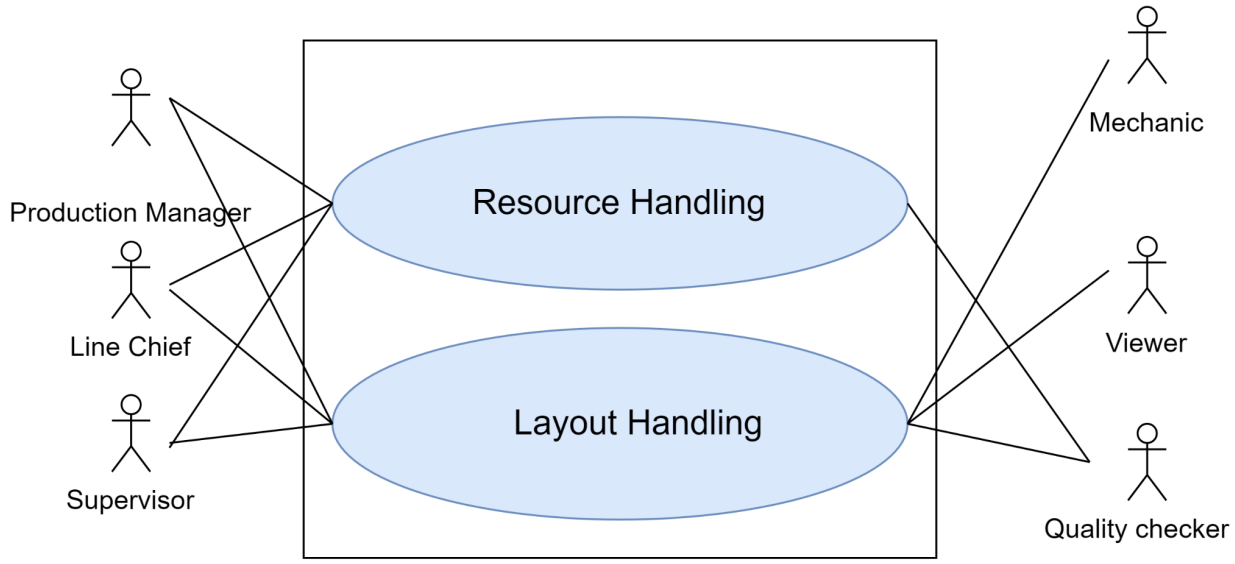
Use case Name: Login
Primary Actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic, viewer, Guest users
Description:
Action reply:
A1: Users log in
R1: System lets the user access the system
A2: Guest users create guest account
R2: System lets the user access the system

Level 1.1.3

Use case Name: Account recovery
Primary Actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic
Description:
Action reply:
A1: users can request for change of password or recover account using password recovery
R1: necessary information is updated

Level 1.2

Use case name: Layout management
Primary actor: Production manager, Line chief, Supervisor
Secondary actor: Viewer, mechanic, Quality checker



L.1.2

Figure 12: Use case diagram level 1.2: Layout management

Description: The whole layout management process has been explicitly explained in this level in two more sub systems. They are Resource handling and layout handling

Level 1.2.1

Use case name: resource handling
Primary actor: Production manager, Line chief, Supervisor
Secondary actor: Quality checker

Action Reply:
A1: Supervisor adds information about requiring resources
R1: line chief is notified
A1: line chief validates request for more resources
R1: Production manager is notified

A1: production manager makes changes in the layout
R1: line details are changed. Supervisor, quality and line chief is notified

Level 1.2.2

Use case name: layout handling
Primary actor: Production manager, Line chief
Secondary actor: Supervisor, Quality checker

Action Reply:
A1: production manager inserts floor information, resources, and assigns line chief
R1: line chief is notified
A1: production manager inserts product design file, and layout information
R1: System generates line index and suggestions

A2: production manager validates and confirms the line
R2: line details are set and viewable to production manager, line chief, supervisors and quality checkers

A3: Production manager sets daily target
R3: Line chief is notified
A3: Line chief sets hourly target
R3: supervisor is notified and everyone can view line

Level 1.3

Use case name: Formal Communication
Primary actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic
Secondary actor: Viewer

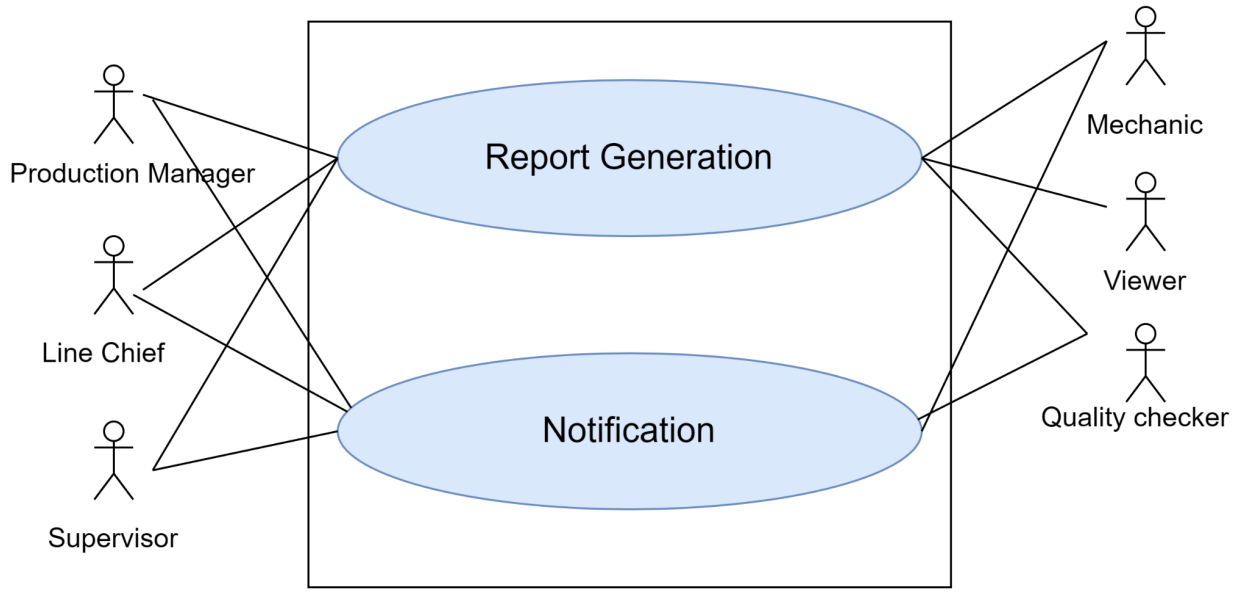


Figure 13: Use case diagram level 1.3: Formal Communication

Description: The whole formal communication process has been explicitly explained in this level in two more sub systems. They are notification and report generation

Note: notification and report generation were two modules that are the main methods of communication in the story. Hence, they are clustered together and named as formal communication module.

Level 1.3.1

Use case name: Report Generation
Primary actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic
Secondary actor: Camera sensor, viewer

Description: Report generation will collect data from all the users, and generate various indexes and reports.
Action Reply:
A1: quality checker inserts hourly production count
R1: supervisor is notified
A1: supervisor validates the count
R1: hourly report is confirmed and uploaded to the system

A2: mechanic checks machine, uploads insight and machine repair details
R2: Machine reports are created and Line chief is notified

A3: quality checkers insert faulty product and backpropagation details
R3: information is stored and supervisor is notified
A3: other index and information is collected
R3: reports generated

Level 1.3.2

Use case name: Notification
Primary actor: Production manager, Line chief, Supervisor, Quality checker, Mechanic

Description:
Action Reply:
A1: sender inserts necessary information and recipient in a specific format
R1: Receiver receives the notification

Level 1.4

Use case name: Congestion Control
Primary actor: Line chief, Supervisor, Quality checker, Mechanic, camera sensor

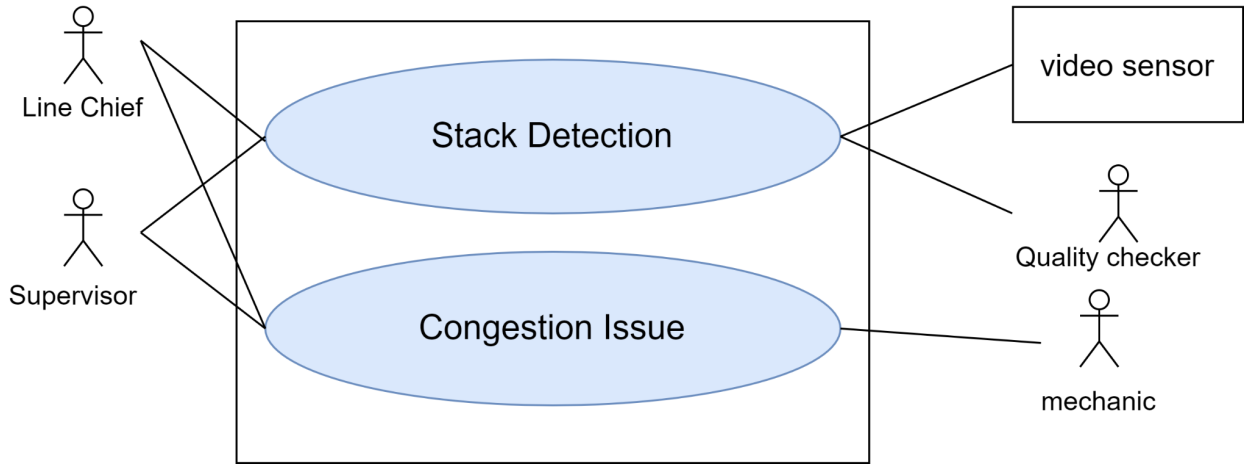


Figure 14: Use case diagram level 1.3: Congestion control

Description: The whole congestion control module has been explicitly explained in this level in two more sub systems. They are stack detection and congestion issues.

Level 1.4.1

Use case name: Stack detection
Primary actor: Line chief, quality checker
Secondary actor: camera sensor, supervisor
Description:
Action Reply:
A1: video sensor detects congestion in the line through stack detection
R1: Line chief is notified
A1: Line chief notifies the supervisor
R1: supervisor checks the line and inserts the reason on the congestion

A2: quality checker report has more errors than tolerance level
R2: line chief is notified
A2: Line chief notifies the supervisor
R2: supervisor checks the line and inserts the reason on the congestion

A3: congestion data stored
R4: congestion report generated

Level 1.4.2

Use case name: congestion issues
Primary actor: Line chief, mechanic, supervisor

Description:
A1: supervisor uploads tutorial to teach workers
R1: workers can view tutorials as guest users

A2: if the reason is machine problems, line chief notifies the mechanic
R2: mechanic accepts or rejects the call

A3: mechanic enters the machine failure cause, inputs whether it is fixable or not
R3: line chief is notified
A3: if machine is not fixable line chief notified production manager for more resources
R3: production manager accepts/rejects
A3: if accepts, supervisor and line chief is notified
R3: line is updated

A4: if the reason is line balancing problem, line chief requests for more resources
R4: production manager accepts/rejects
A4: if accepts, supervisor and line chief is notified
R4: line is updated

4.3 Activity Diagram

Definition of Activity Diagram: Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency.

Activity diagram ID: 01

Level 1.1.1
Name: Register account
Reference: Use case level 1.1 (Figure 11: use case diagram level 1.1 user management)

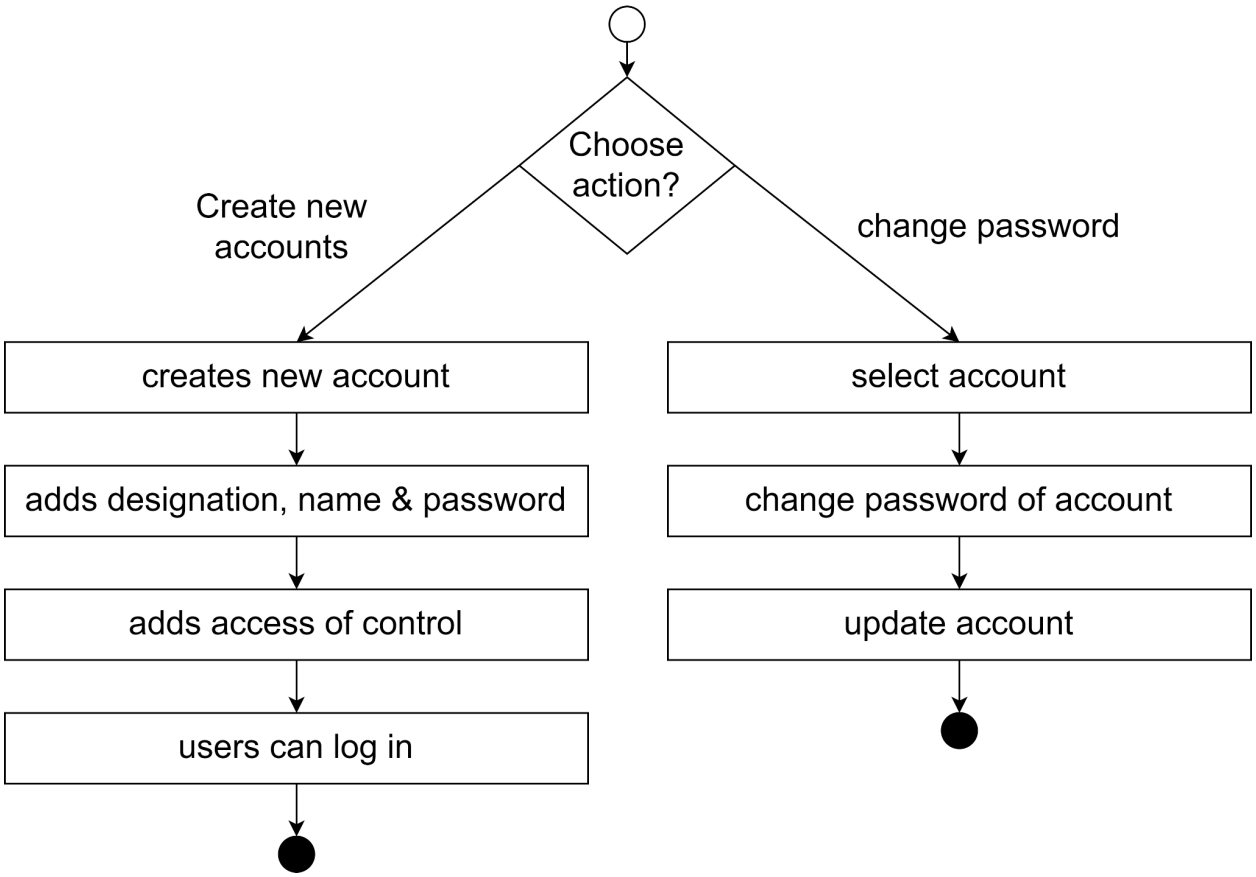


Figure 15: activity diagram level 1.1.1 register account

Activity diagram ID: 02

Level 1.1.2

Name: Login

Reference: Use case level 1.1 (Figure 11: use case diagram level 1.1 user management)

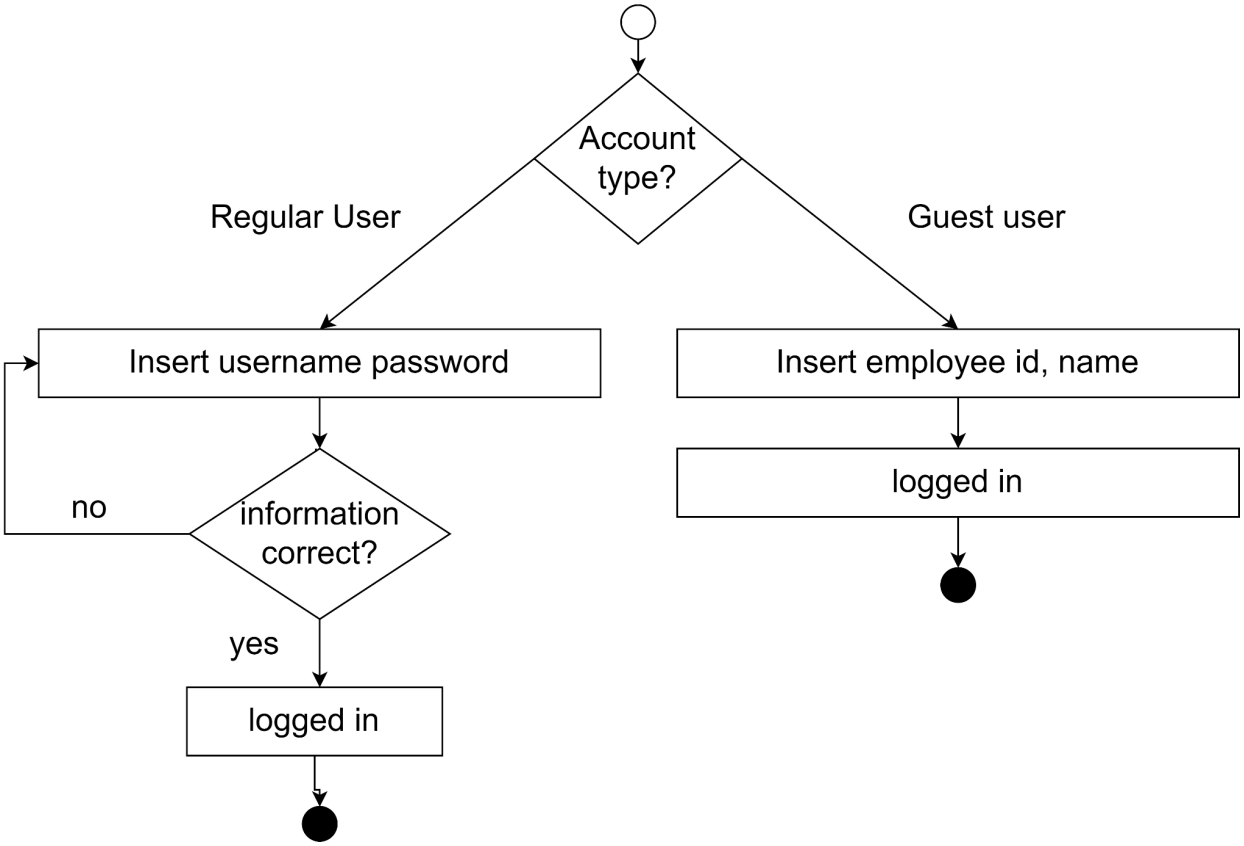


Figure 16: activity diagram level 1.1.2 login

Activity diagram ID: 03

Level 1.1.3

Name: Account information change
Reference: Use case level 1.1 (Figure 11: use case diagram level 1.1 user management)

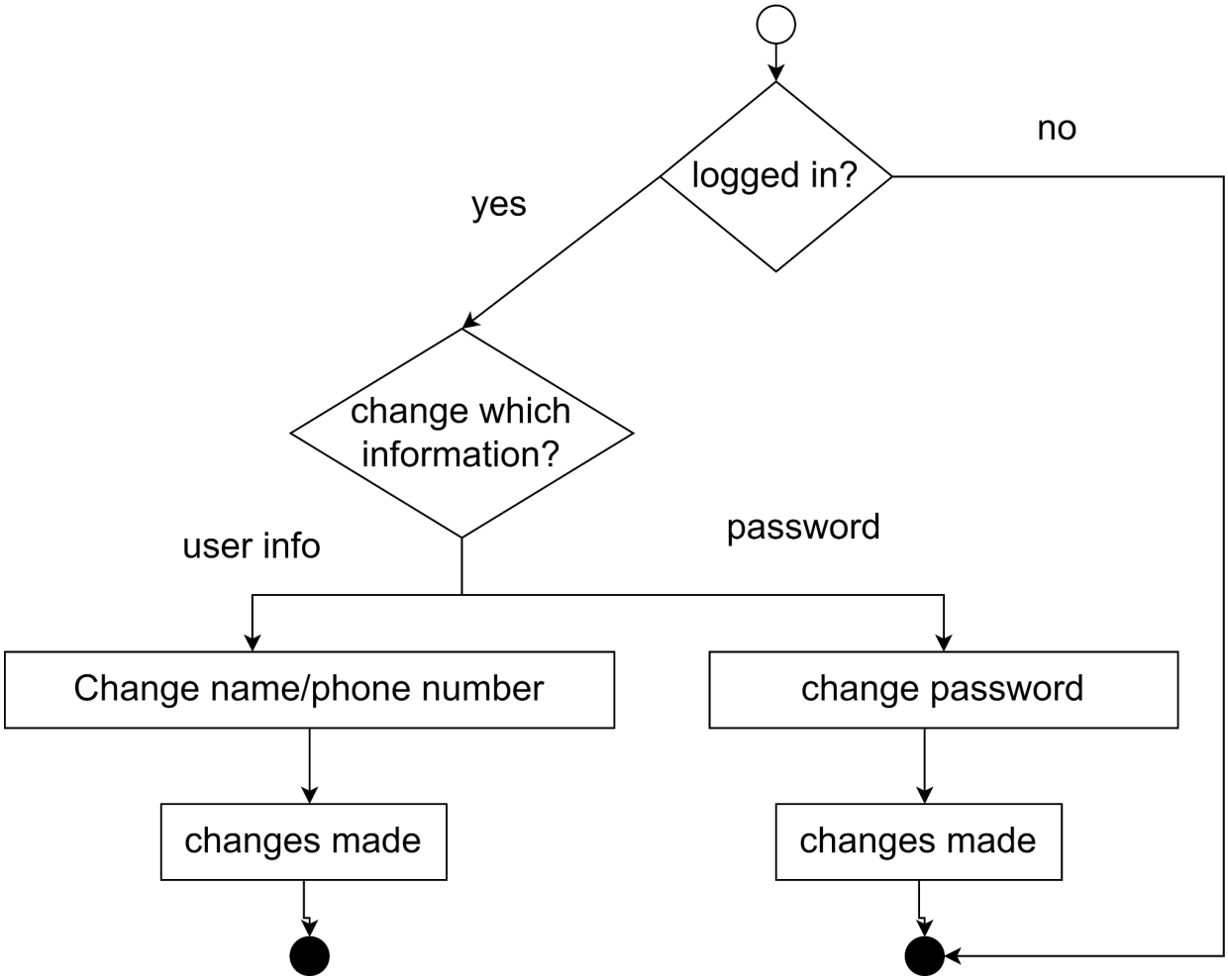


Figure 17: activity diagram level 1.1.3 Account information change

Activity diagram ID: 04

Level 1.2.1

Name: resource handling
Reference: Use case level 1.2 (Figure 12: Use case diagram level 1.2: Layout management)

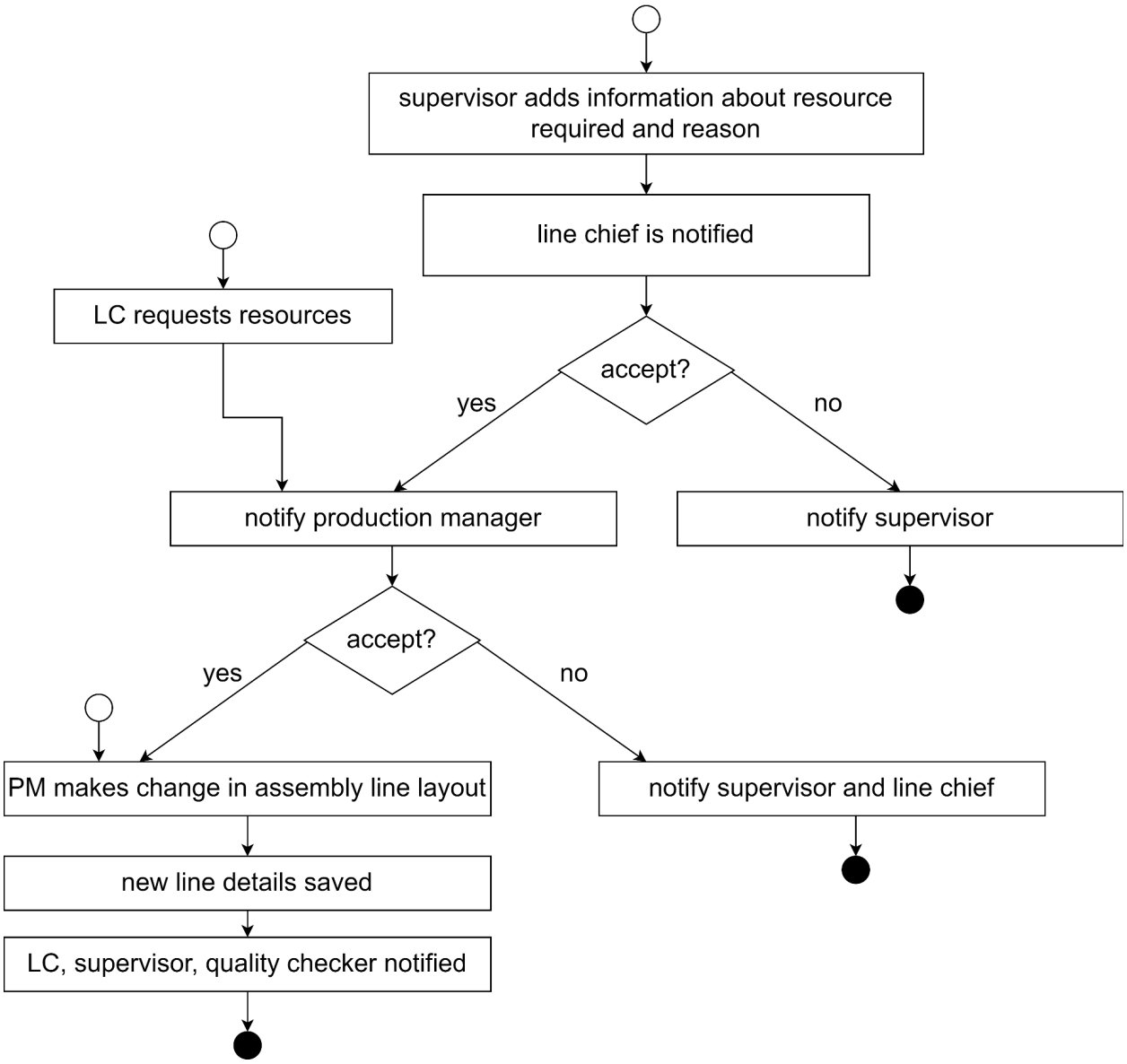


Figure 18: activity diagram level 1.2.1 resource handling

Activity diagram ID: 05

Level 1.2.2

Name: layout handling

Reference: Use case level 1.2 (Figure 12: Use case diagram level 1.2: Layout management)

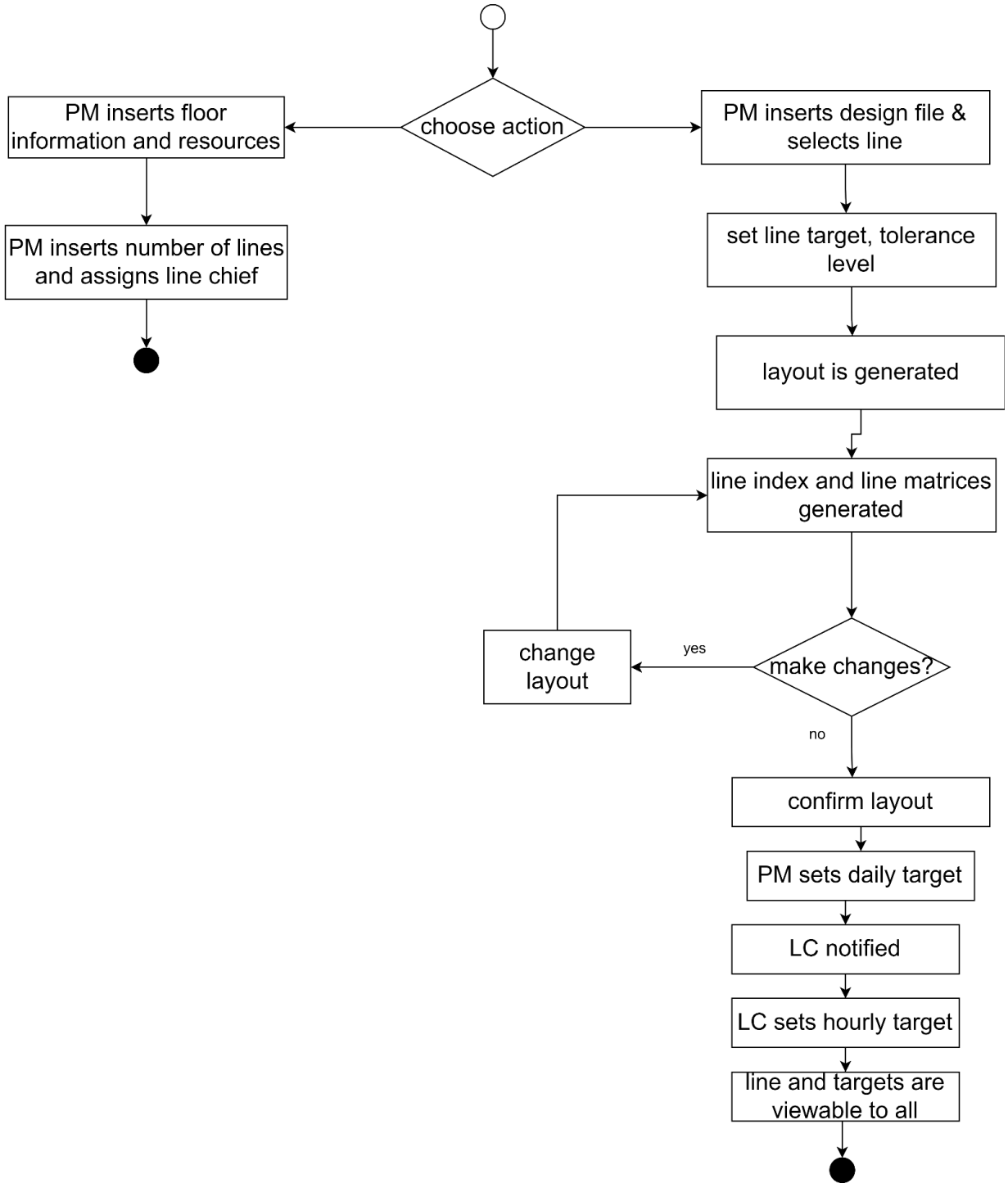


Figure 19: activity diagram level 1.2.2 layout handling

Activity diagram ID: 06

Level 1.3.1

Name: report generation
Reference: Use case level 1.3 (Figure 13: Use case diagram level 1.3: Formal Communication)

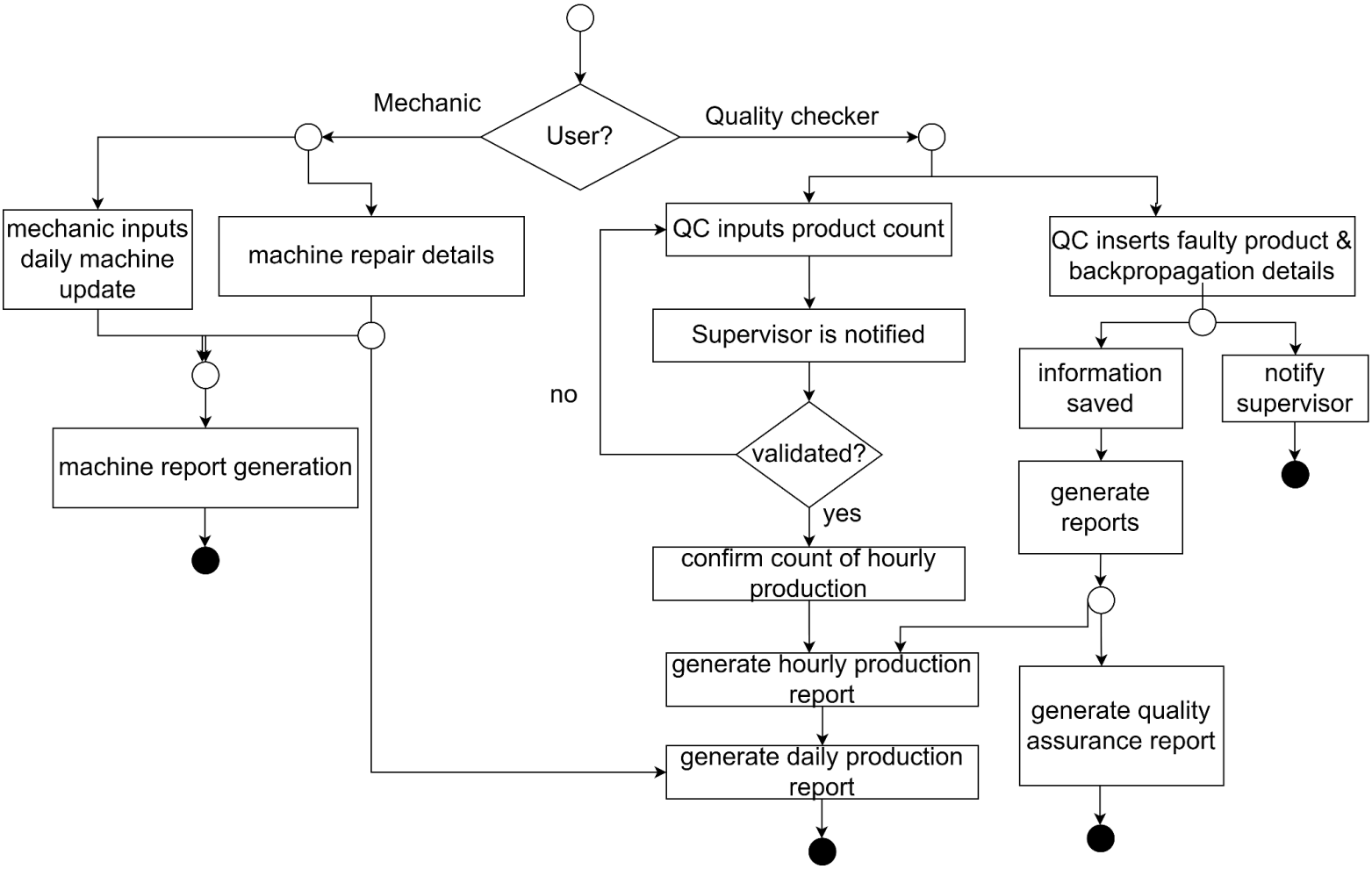


Figure 20: activity diagram level 1.3.1 report generation

Activity diagram ID: 07

Level 1.3.2
Name: notification
Reference: Use case level 1.3 (Figure 13: Use case diagram level 1.3: Formal Communication)

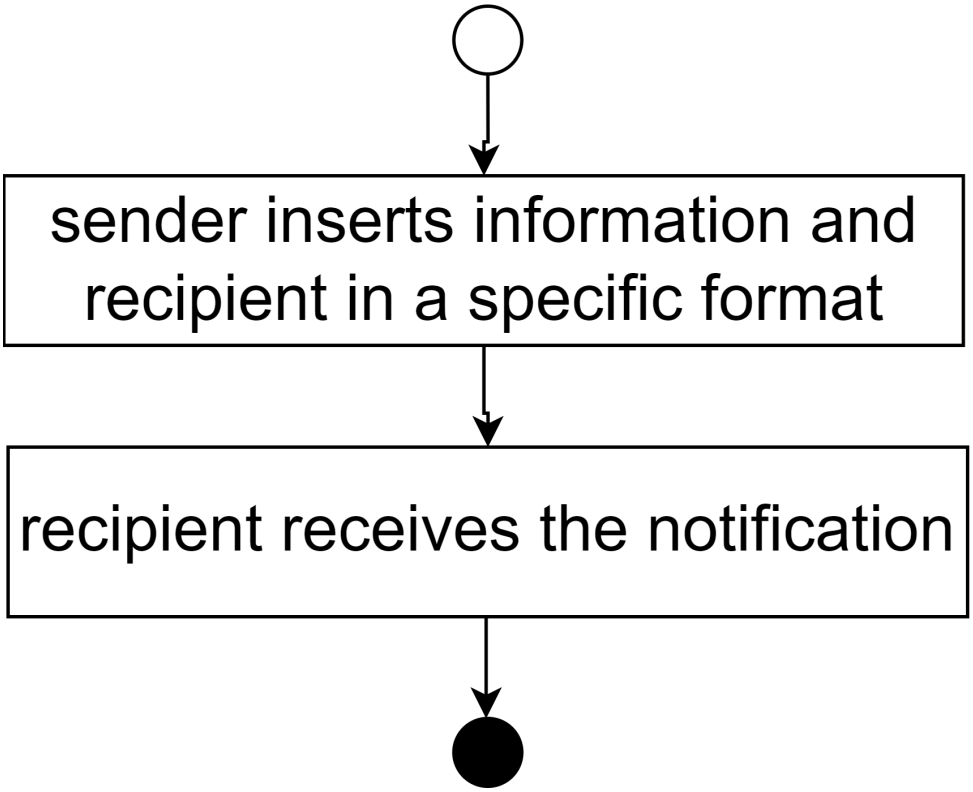


Figure 21: activity diagram level 1.3.2 notification

Activity diagram ID: 08

Level 1.4.1

Name: stack detection

Reference: Use case level 1.4 (Figure 14: Use case diagram level 1.4: Congestion control)

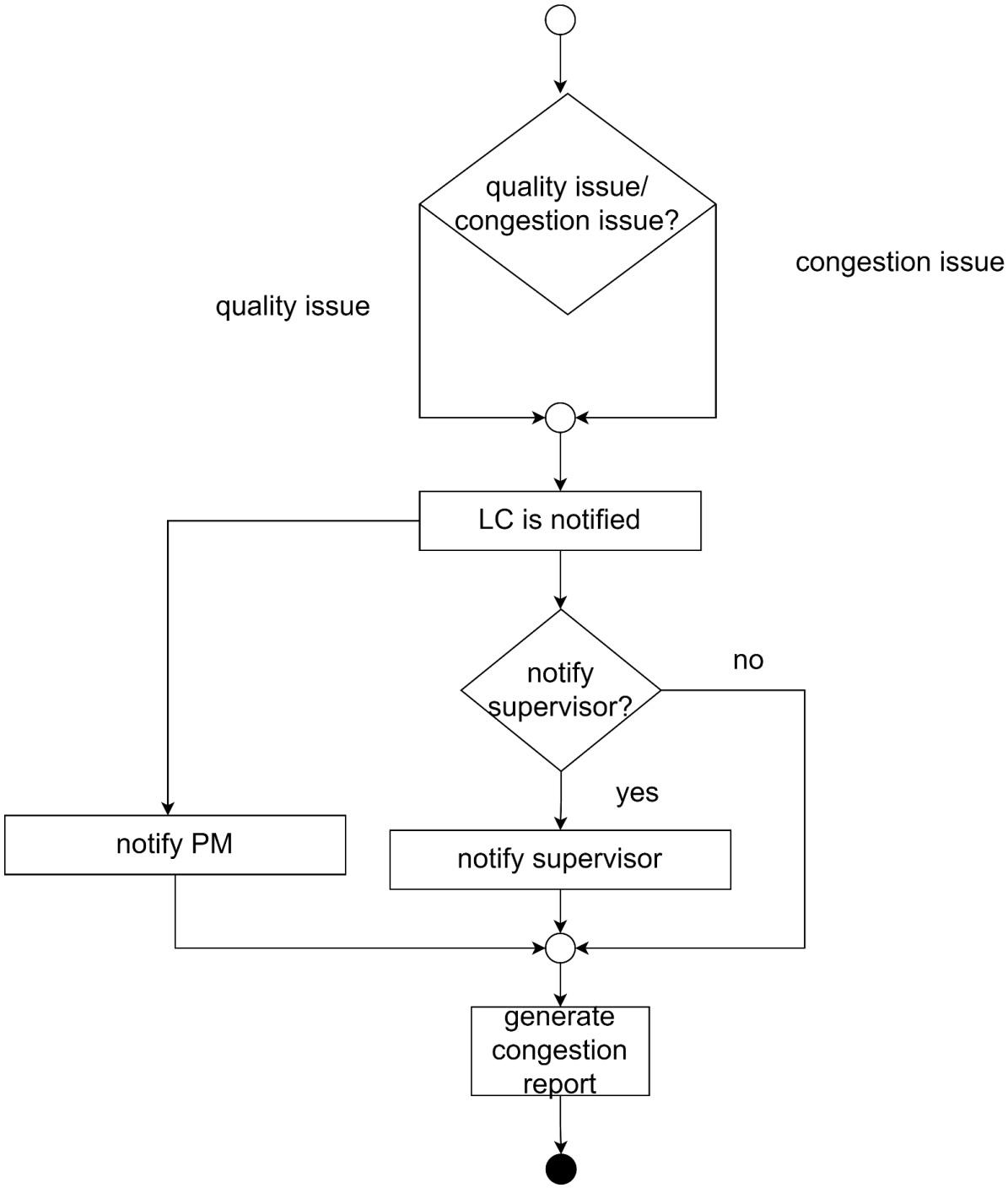


Figure 22: activity diagram level 1.4.1 stack detection

Activity diagram ID: 09

Level 1.4.2

Name: Congestion issues

Reference: Use case level 1.4 (Figure 14: Use case diagram level 1.4: Congestion control)

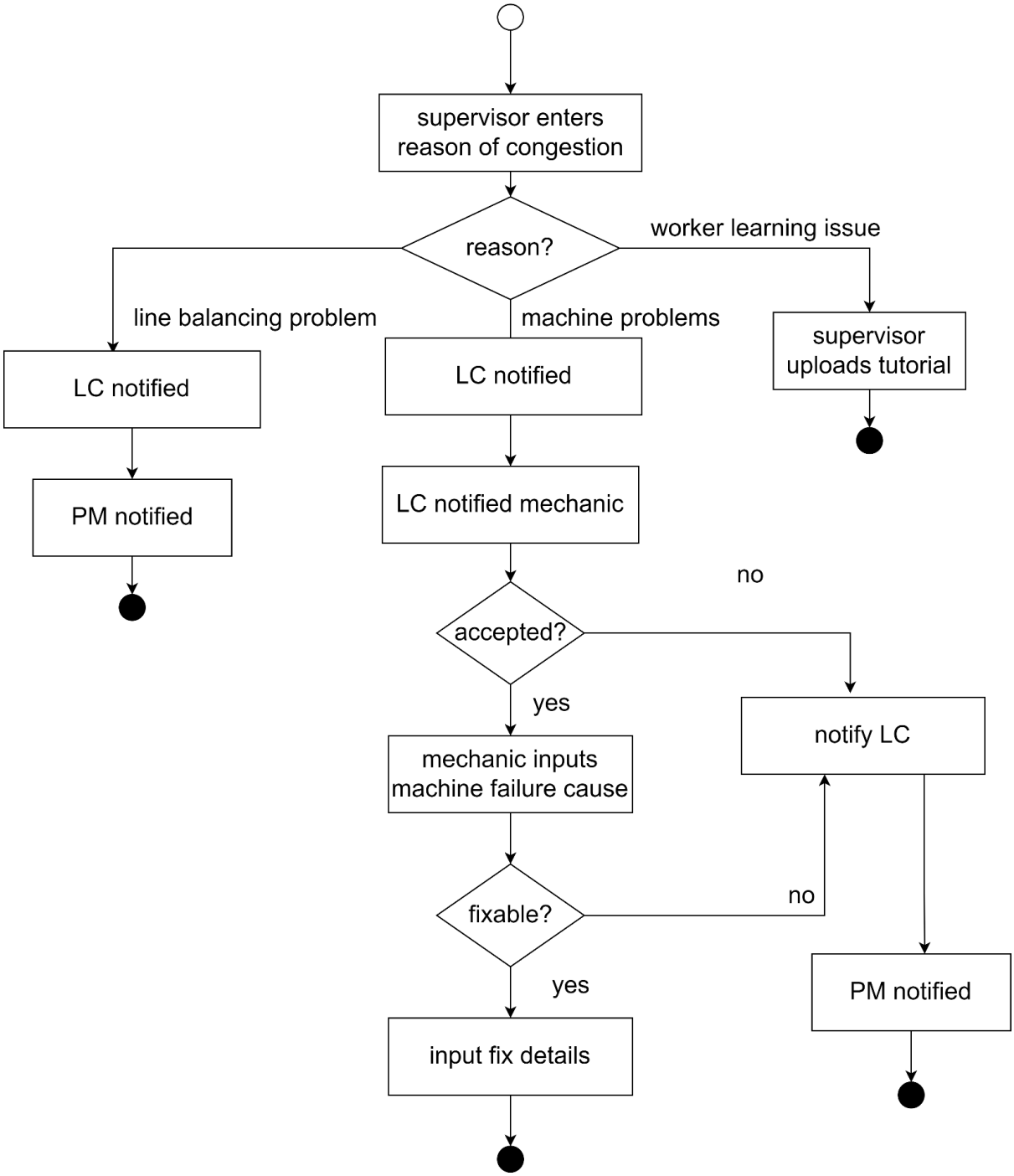


Figure 23: activity diagram level 1.4.2 Congestion issues

4.4 Swim lane diagrams

A swimlane diagram is a type of flowchart. Like a flowchart, it diagrams a process from start to finish, but it also divides these steps into categories to help distinguish which departments or employees are responsible for each set of actions. It is based on the analogy of lanes in a pool, as it places process steps within the horizontal or vertical “swimlanes” of a particular department, work group or employee, thus ensuring clarity and accountability.

Swim lane Diagram ID: 01

Level 1.1.1

Name: Register account
Reference: Use case level 1.1 (Figure 11: use case diagram level 1.1 user management) & activity diagram ID: 01 (Figure 15: activity diagram level 1.1.1 register account)

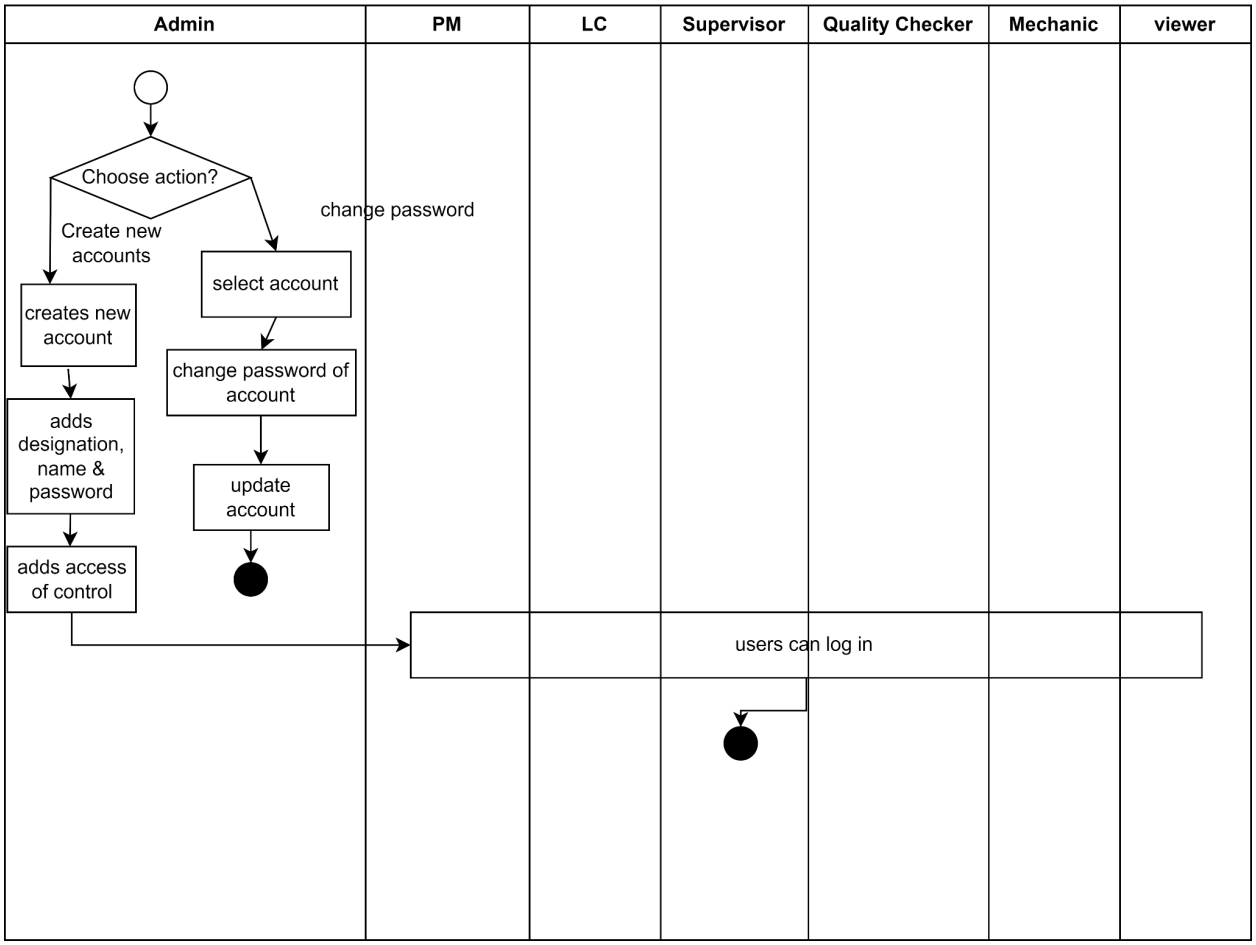


Figure 24: swimlane diagram level 1.1.1 register account

Swimlane diagram ID: 02

Level 1.1.2

Name: Login

Reference: Use case level 1.1 (Figure 11: use case diagram level 1.1 user management) & activity diagram ID: 02 (Figure 16: activity diagram level 1.1.2 login)

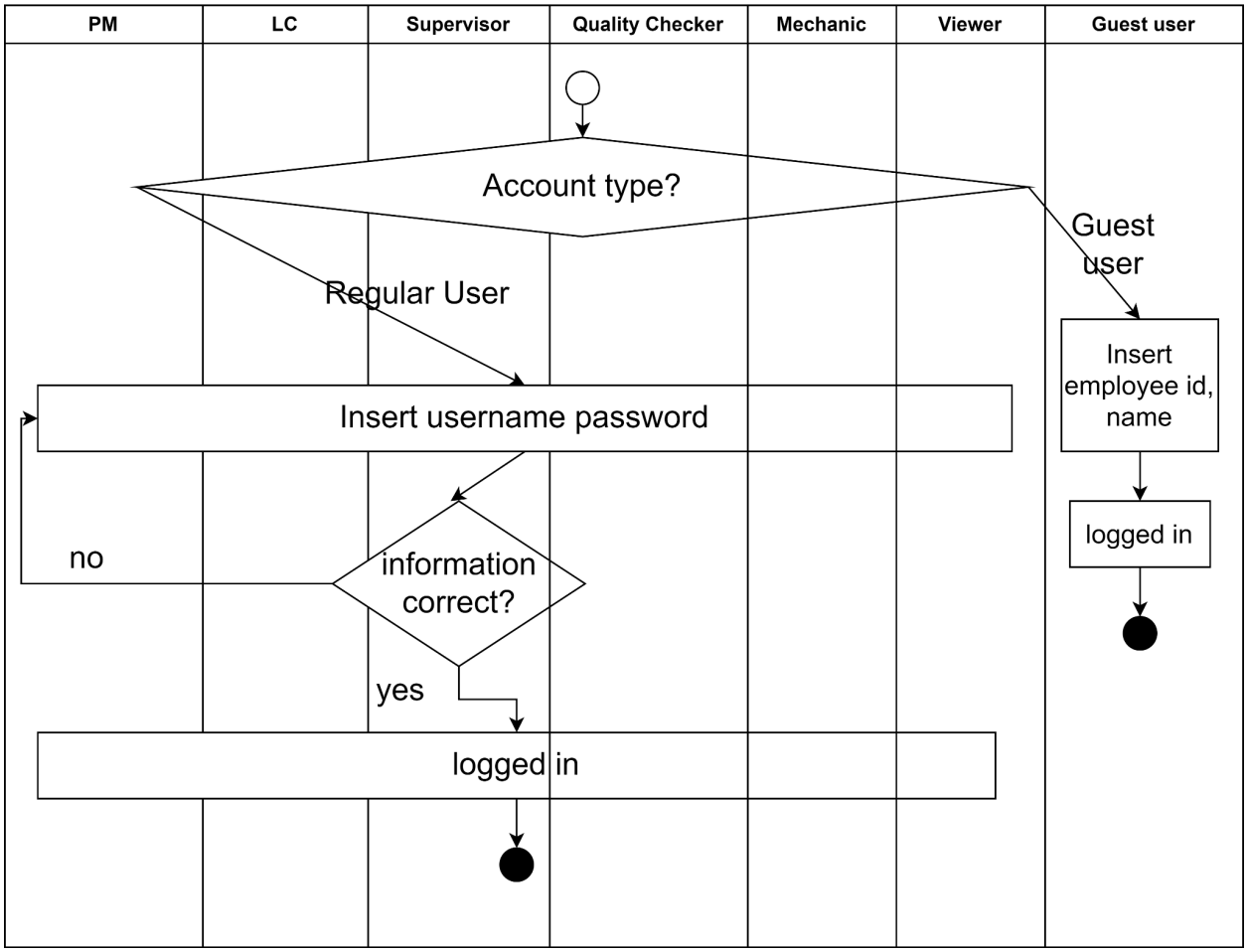


Figure 25: swimlane diagram level 1.1.2 login

Swimlane diagram ID: 03

Level 1.1.3

Name: Account information change
Reference: Use case level 1.1 (Figure 11: use case diagram level 1.1 user management) & Activity diagram ID: 03 (Figure 17: activity diagram level 1.1.3 Account information change)

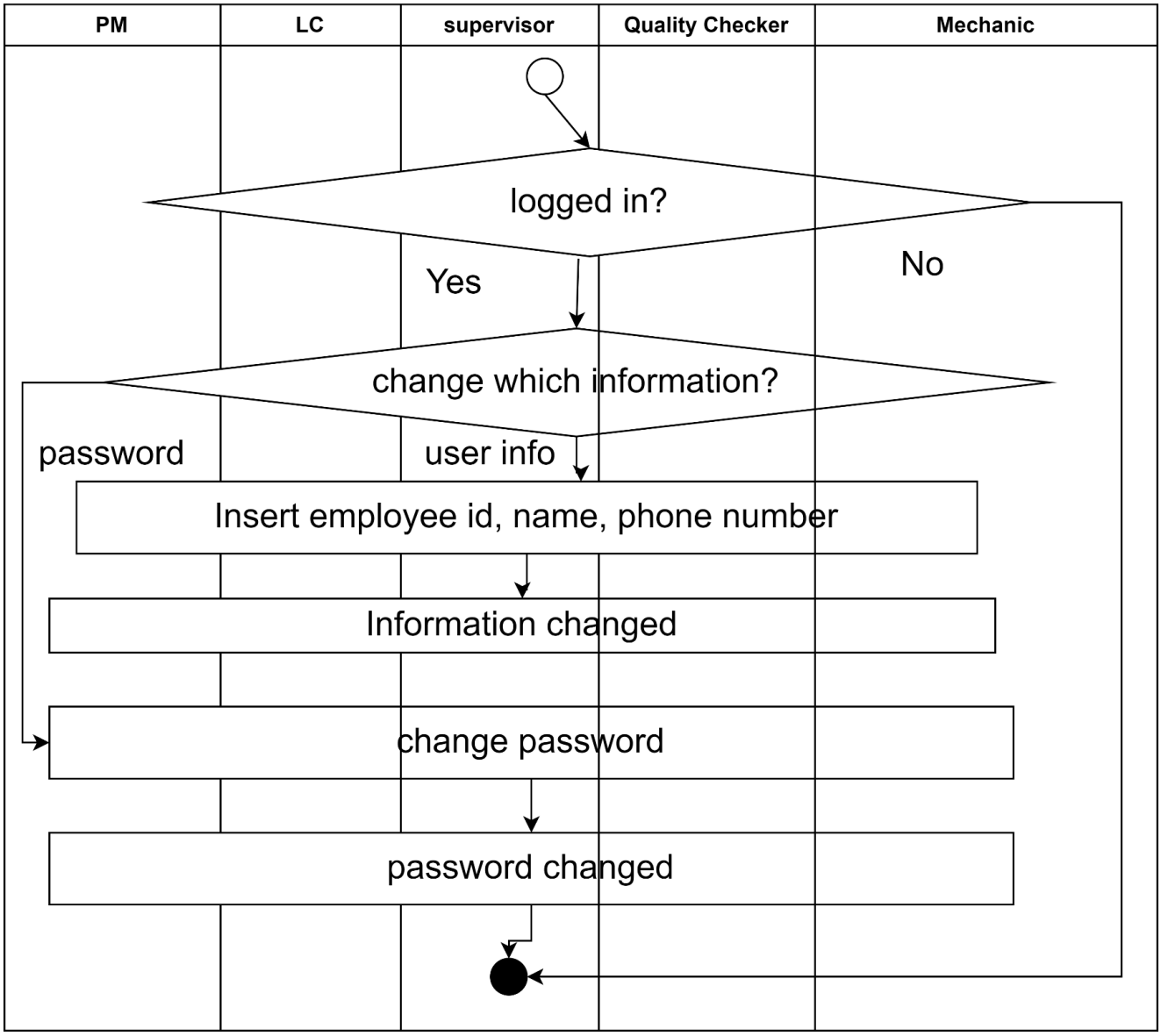


Figure 26: swimlane diagram level 1.1.3 Account information change

Swimlane diagram ID: 04

Level 1.2.1

Name: resource handling

Reference: Use case level 1.2 (Figure 12: Use case diagram level 1.2: Layout management) & Activity diagram ID: 04 (Figure 18: activity diagram level 1.2.1 resource handling)

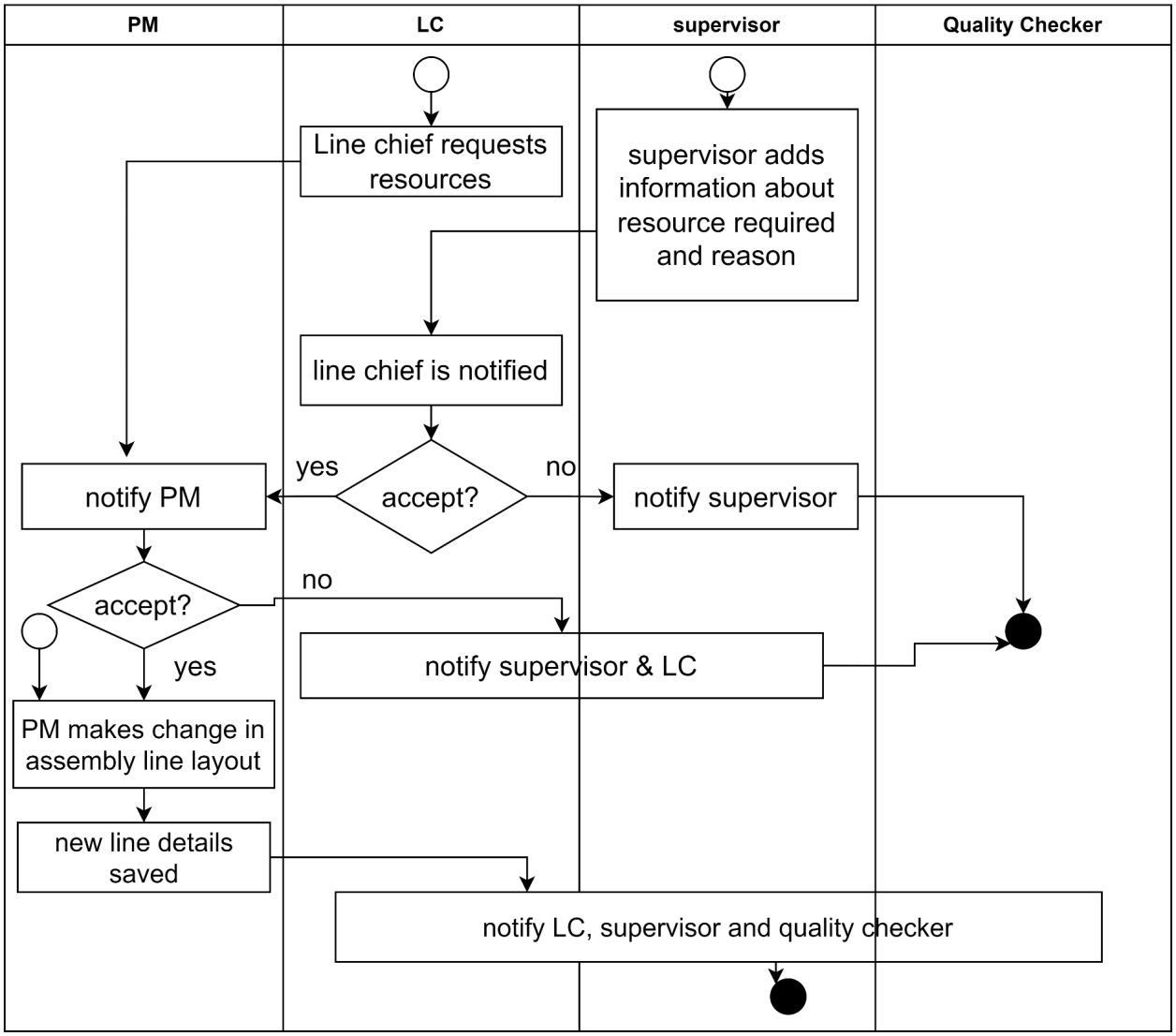


Figure 27: swimlane diagram level 1.2.1 resource handling

Swimlane diagram ID: 05

Level 1.2.2

Name: layout handling

Reference: Use case level 1.2 (Figure 12: Use case diagram level 1.2: Layout management) & Activity Diagram ID:05 (Figure 19: activity diagram level 1.2.2 layout handling)

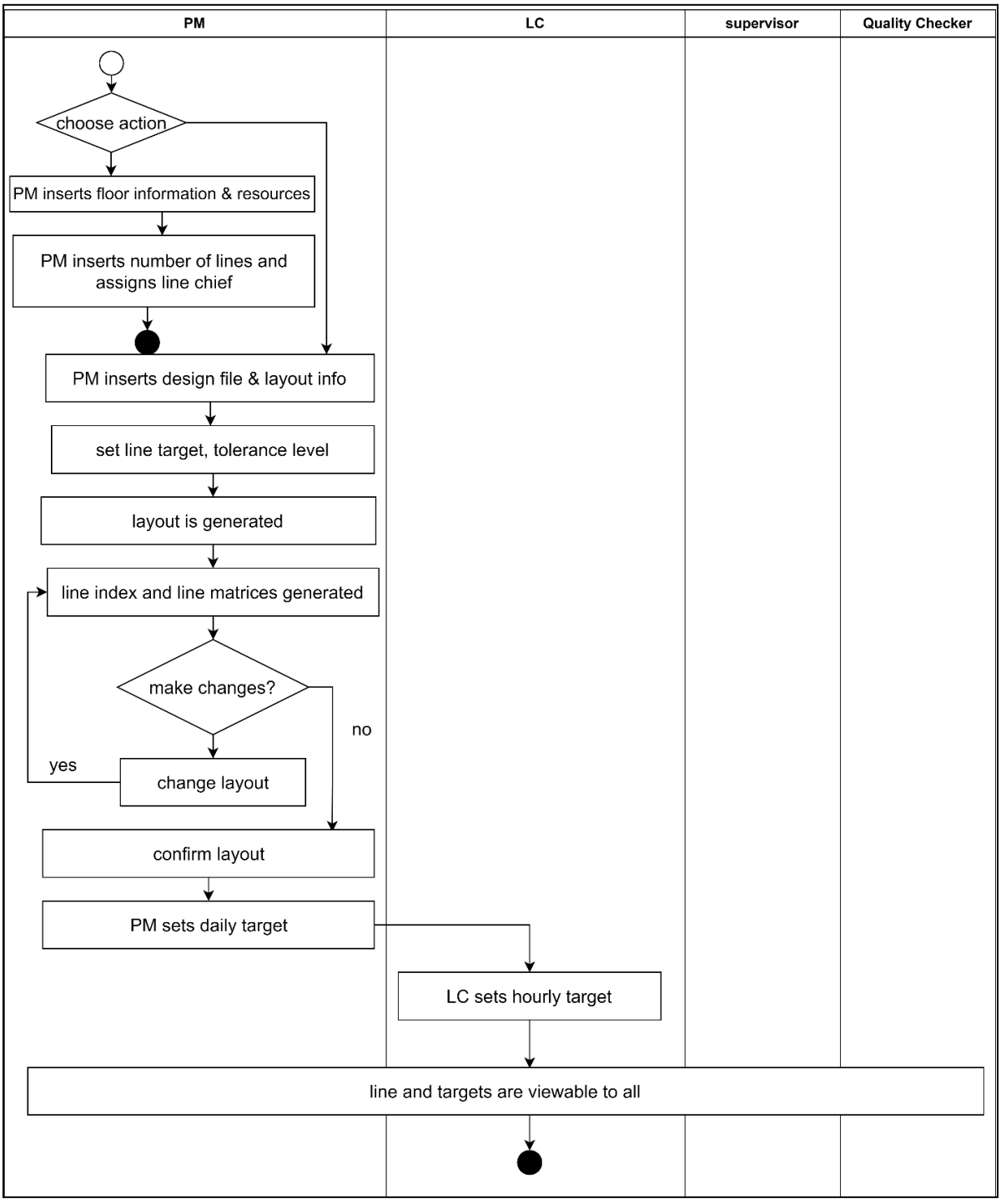


Figure 28: swimlane diagram level 1.2.2 layout handling

Swimlane diagram ID: 06

Level 1.3.1

Name: report generation

Reference: Use case level 1.3 (Figure 13: Use case diagram level 1.3: Formal Communication) & activity diagram ID:06 (Figure 20: activity diagram level 1.3.1 report generation)

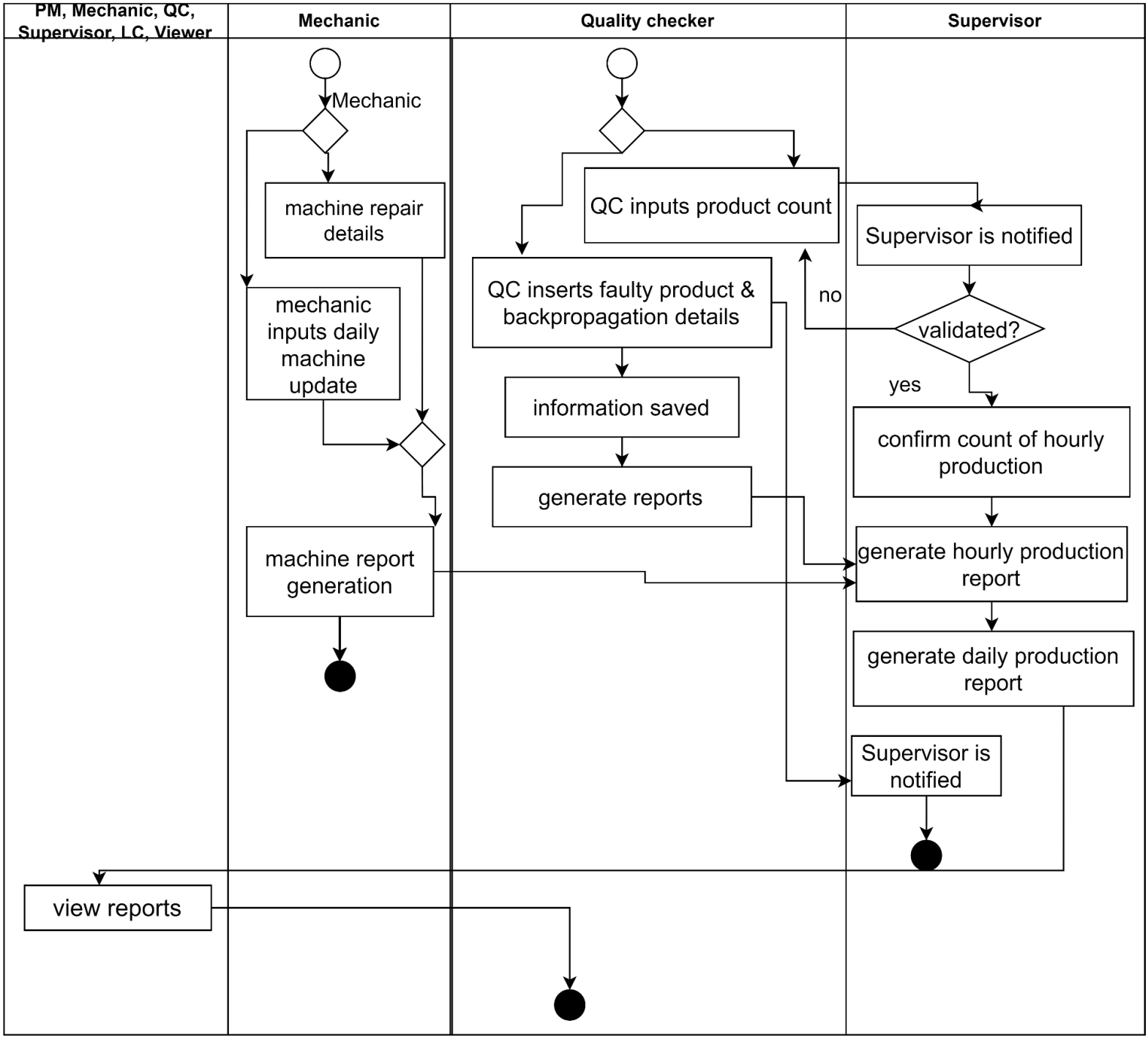


Figure 29: swimlane diagram level 1.3.1 report generation

Swimlane diagram ID: 07

Level 1.3.2

Name: notification
Reference: Use case level 1.3 (Figure 13: Use case diagram level 1.3: Formal Communication) & activity diagram ID: 07 (Figure 21: activity diagram level 1.3.2 notification)

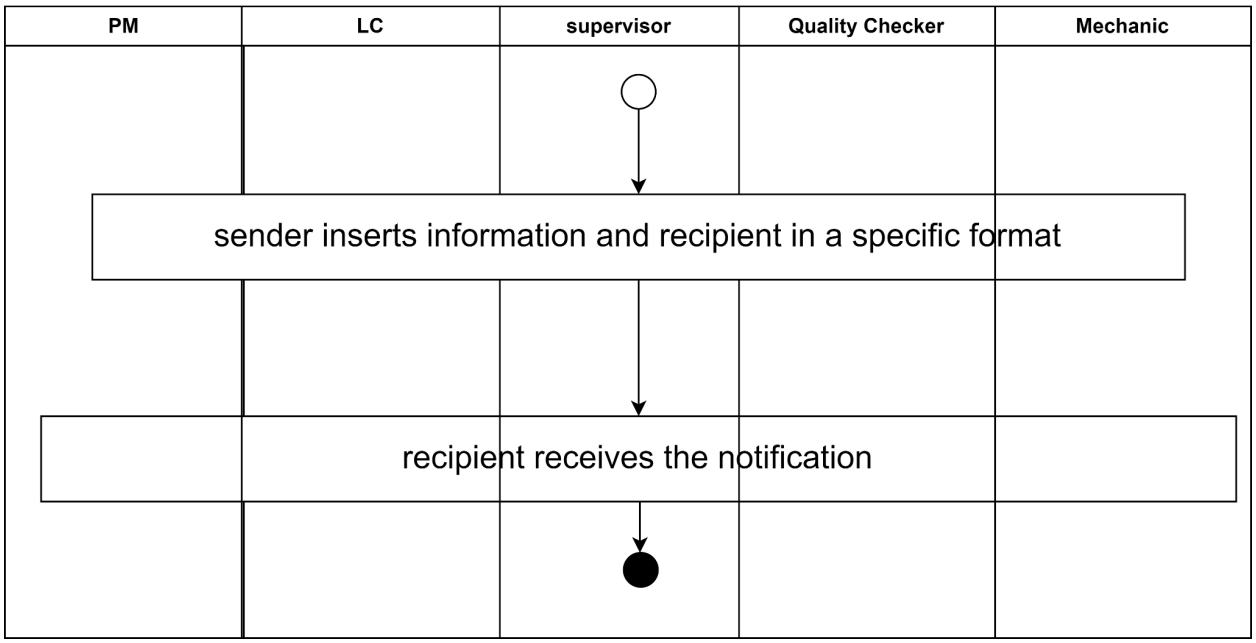


Figure 30: swimlane diagram level 1.3.2 notification

Swimlane diagram ID: 08

Level 1.4.1

Name: stack detection

Reference: Use case level 1.4 (Figure 14: Use case diagram level 1.4: Congestion control) & activity diagram ID: 08 (Figure 22: activity diagram level 1.4.1 stack detection)

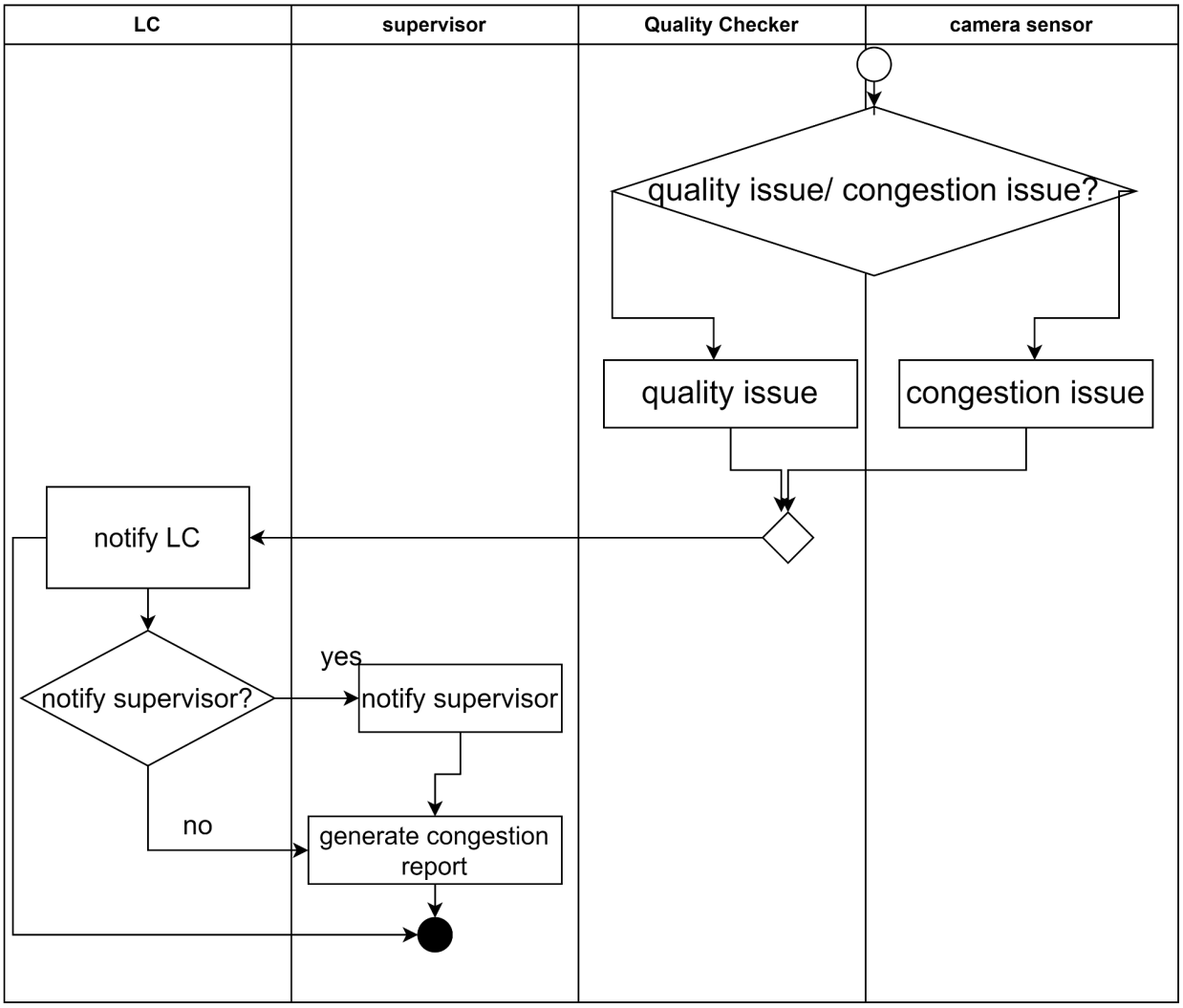


Figure 31: swimlane diagram level 1.4.1 stack detection

Swimlane diagram ID: 09

Level 1.4.2

Name: Congestion issues

Reference: Use case level 1.4 (Figure 14: Use case diagram level 1.4: Congestion control) & activity diagram ID:09 (Figure 23: activity diagram level 1.4.2 Congestion issues)

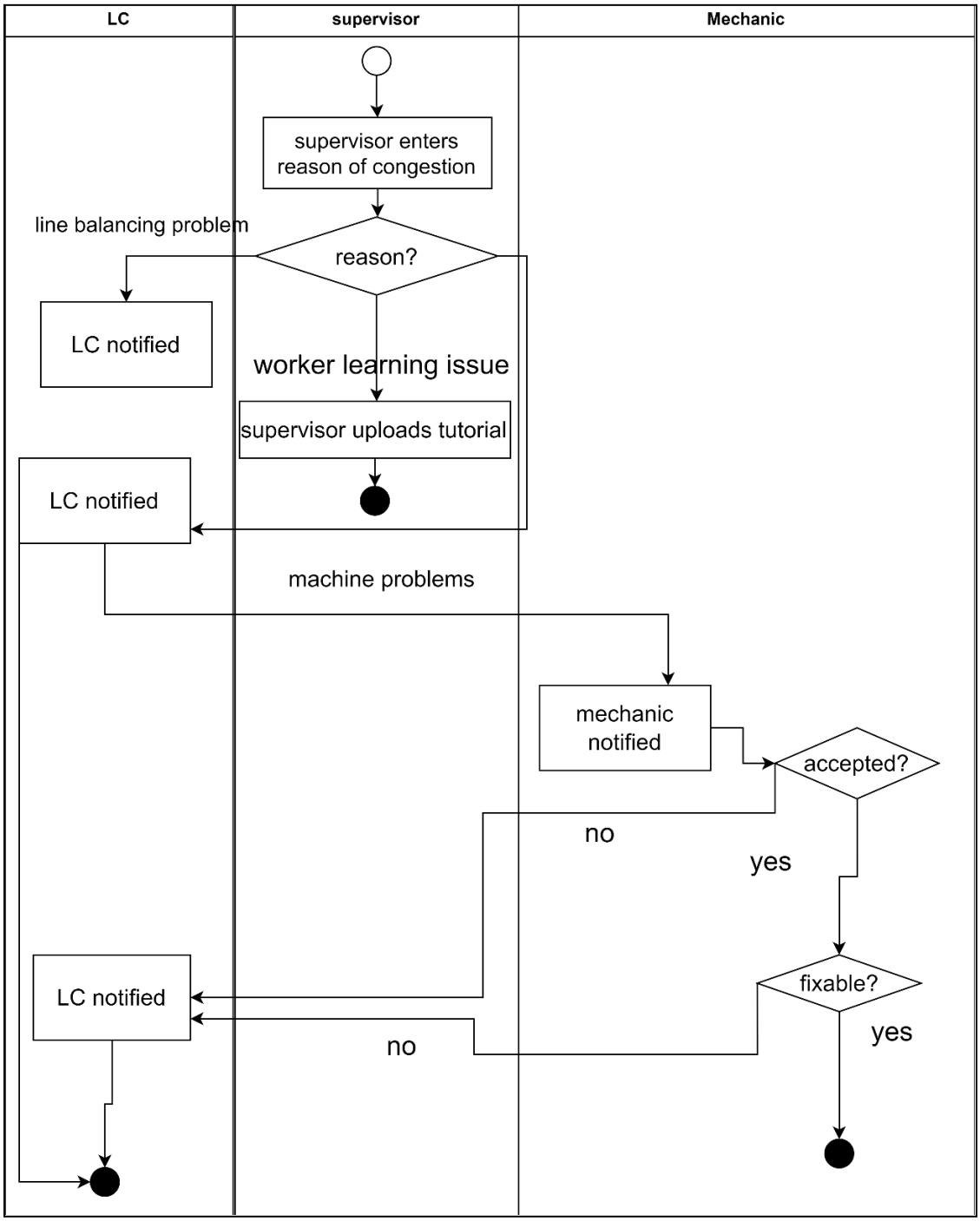


Figure 32: swimlane diagram level 1.4.2 Congestion issues

5. Class Based Modeling

Class-based modeling defines the structure of the entire system by identifying the static structure of objects in that system. A class model defines attributes and operations for the objects of each class and also the relationship between the objects, and the collaborations that occur between the classes of the systems. The elements of a class-based model include classes and objects, attributes, operations, class-responsibility-collaborator (CRC) models, collaboration diagrams, and packages.

5.1 Noun List

We follow a grammatical approach to find the classes and its elements, attributes. The reason being an inefficient dependability on novice intuition. The grammatical approach requires us to extract the nouns first from the story.

5.1.1 Problem Space

Using a POS-tagger, we have identified 130 nouns.

Noun	Noun	Noun
Admin user	Total number of Productions vs Target	Tutorials
Default user	Issues that happened throughout the day	Workers
System	Quality Index	Guest User
Users	Machine Issues Index	System
Production Manager	Required Overtime	Issues
Mechanic	Number of Production Deficit	Process
Supervisor	Hourly Production Rate	Assembly Line Stacks
Quality Checker	Live Layout	Stack Height
Viewer	Target per Workstation	Limit
Information	Quality Issues per Workstation	Congestion

Resources	Assembly Line Performance Matrices	Congested Workstation
Overlock sewing machine	Target vs Achieved production	Congestion
Assembly Table	Live Congestion Results	Reason for Congestion
Marking Table	Hourly Completed Product	Case
Electronic Sewing Machine	Hourly Rejected Product	Worker ’s Learning Issues
Number of Assembly Lines	Start time	New Task
Control	End time	Machine Issues
Types of Machine	Average Cycle Time	Request
Assembly Line	Daily Production Report	Problem
Design File	Production per hour of the day	Machine Failure Issue
Product	Target Production	New Workstation
Types of Tasks	Reached Production	Available Resources
Types of Machines	Issues	Line Balancing Problems
Number of Machines	Fall of Production	Requests Workstations
Sizes of the Cutting Pieces	Allocated Workstation	Over-workstation Issues
Size of the Final Product	Quality Assurance Report	Line Balancing
Number of Products	Causes of Quality Fall	Issues
Assembly Line Layout	Assembly line graph of quality drop	Workstation Fixing Information
Interactive Layout Interface	Machine Report	Machine Reports
Available Information	Report	End of the Assembly Line
Matrices	Number of Broken machines	Number of Accepted Products
Line efficiency	Visual Sensor	Number of Rejected Products
Resource Utilization Index	Number of Fixed machines	Hourly Production Report

Possible Bottleneck points	Machine status	Backpropagation Information
Notification	Time Wasted	Dynamic Line Balancing
Line Chief	Congestion Report	Load
Targets	Congestion graph of the assembly line layout	Quality
Tolerance Level of Error	Congestion count	Issue of Quality Drop
Hourly Targets	Efficiency vs Congestion graph	Counting
Workstation	Line Balancing Report	Approval
Production	Bottleneck points graph	Allocation of Resources
Day	Time Series Graph	End of the Workday
Piece of Product	Monthly Assembly line performance report	Production report

5.1.2 Solution Space

We have a total of 130 nouns in problem space. After elimination through intuition, we have 96 nouns in solution space.

Noun	Solution Space	Used?	Noun	Solution Space	Used?
Admin user			New Workstation		
Default user	X	n	Available Resources		
System	X	n	Line Balancing Problems		
Users			Requests Workstations		

Production Manager			Over-workstation Issues		
Mechanic			Line Balancing		
Supervisor			Issues	X	
Quality Checker			Workstation Fixing Information		
Viewer			Machine Reports		
Information	X	n	End of the Assembly Line	X	n
Resources			Number of Accepted Products		
Overlock sewing machine	X		Number of Rejected Products		
Assembly Table	X		Hourly Production Report		
Marking Table	X		Backpropagation Information		
Electronic Sewing Machine	X		Dynamic Line Balancing		
Number of Assembly Lines			Load	X	n
Control	X	n	Quality		
Types of Machine	X		Issue of Quality Drop		

Assembly Line			Counting	X	
Design File			Approval		
Product			Allocation of Resources		
Types of Tasks	X		End of the Workday	X	n
Types of Machines			Production report		
Number of Machines			Total number of Productions vs Target	X	
Sizes of the Cutting Pieces			Issues that happened throughout the day	X	
Size of the Final Product	X		Quality Index		
Number of Products			Machine Issues Index		
Assembly Line Layout			Required Overtime		
Interactive Layout Interface			Number of Production Deficit		
Available Information			Hourly Production Rate	X	
Matrices			Live Layout		
Line efficiency			Target per Workstation		
Resource Utilization Index			Quality Issues per Workstation		

Possible Bottleneck points			Assembly Line Performance Matrices		
Notification			Target vs Achieved production	X	
Line Chief			Live Congestion Results		
Targets			Hourly Completed Product		
Tolerance Level of Error			Hourly Rejected Product		
Hourly Targets			Start time	X	n
Workstation	X		End time	X	n
Production			Average Cycle Time		
Day			Daily Production Report		
Piece of Product	X		Production per hour of the day		
Tutorials			Target Production		
Workers	X	n	Reached Production		
Guest User			Issues		
System	X	n	Fall of Production	X	n
Issues			Allocated Workstation	X	

Process	X	n	Quality Assurance Report		
Assembly Line Stacks			Causes of Quality Fall	X	
Stack Height			Assembly line graph of quality drop		
Limit	X		Machine Report		
Congestion			Report		
Congested Workstation			Number of Broken machines		
Congestion	X		Visual Sensor		
Reason for Congestion			Number of Fixed machines		
Case	X	n	Machine status		
Worker ’s Learning Issues			Time Wasted		
New Task	X	n	Congestion Report		
Machine Issues			Congestion graph of the assembly line layout		
Request			Congestion count	X	
Problem	X		Efficiency vs Congestion graph		

Machine Failure Issue	X		Line Balancing Report		
			Bottleneck points graph		
			Time Series Graph		
			Monthly Assembly line performance report		

5.1.3 General Characteristics (GC)

We followed a generous scoring ideal when assigning the score. The reason is - It is more harmful to lose a potential class due to harsh scoring criteria than having more classes than necessary due to generous scoring. Because we can just analyze the class and merge them to address this issue.

The general characteristics are:

1. External entities (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
2. Things (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. Occurrences or events (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. Roles (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. Organizational units (e.g., division, group, team) that are relevant to an application.
6. Places (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. Structures (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects

Noun	**GC Score**
Admin user	4,5,7

Users	4,5,7
Production Manager	4,5,7
Mechanic	4,5,7
Supervisor	4,5,7
Quality Checker	4,5,7
Viewer	4,5,7
Resources	2,3,4
Visual Sensor	***1***
Number of Assembly Lines	2
Assembly Line	3,5,6
Design File	2,7
Product	2,3,5,7
Report	2,3,5
Types of Machines	2
Number of Machines	2
Sizes of the Cutting Pieces	2
Number of Products	2
Assembly Line Layout	3,5,6,7
Interactive Layout Interface	3,5,6,7
Available Information	2
Matrices	2,3,7
Line efficiency	2
Resource Utilization Index	2
Possible Bottleneck points	2
Notification	2,3
Line Chief	4,5,7
Targets	2
Tolerance Level of Error	2
Hourly Targets	2
Production	3,6,7

Day	2
Tutorials	2
Guest User	4,5,7
Issues	2,3,6
Assembly Line Stacks	2
Stack Height	2
Congestion	3,5,7
Congested Workstation	2
Reason for Congestion	2
Worker ’s Learning Issues	2
Machine Issues	2
Request	3,5,6
New Workstation	3
Available Resources	2
Line Balancing Problems	3
Requests Workstations	3
Over-workstation Issues	3
Line Balancing	3,6,7
Workstation Fixing Information	2
Machine Reports	2,3
Number of Accepted Products	2
Number of Rejected Products	2
Hourly Production Report	2
Backpropagation Information	2
Dynamic Line Balancing	3,5,6,7
Quality	3,5,7
Issue of Quality Drop	2,3
Approval	3
Allocation of Resources	3
Production report	2

Quality Index	2
Machine Issues Index	2
Required Overtime	2,3
Number of Production Deficit	2
Live Layout	3,5,6
Target per Workstation	2
Quality Issues per Workstation	2,3
Assembly Line Performance Matrices	2
Live Congestion Results	2
Hourly Completed Product	2
Hourly Rejected Product	2
Average Cycle Time	2
Daily Production Report	2
Production per hour of the day	2
Target Production	2
Reached Production	2
Issues	3,5,6,7
Quality Assurance Report	2
Assembly line graph of quality drop	2
Machine Report	2,3
Number of Broken machines	2
Number of Fixed machines	2
Machine status	2
Time Wasted	2
Congestion Report	2,3
Congestion graph of the assembly line layout	2
Efficiency vs Congestion graph	2
Line Balancing Report	2,3

Bottleneck points graph	2
Time Series Graph	2
Monthly Assembly line performance report	2,3

5.1.4 Selection Criteria

Coad and Yourdon [Coa91] suggest six selection characteristics that should be used as you consider each potential class for inclusion in the analysis model:

1. **Retained information:** The potential class will be useful during analysis only if information about it must be remembered so that the system can function.
2. **Needed services**: The potential class must have a set of identifiable operations that can change the value of its attributes in some way.
3. **Multiple attributes:** During requirement analysis, the focus should be on "major" information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.
4. **Common attributes**: A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
5. **Common operations**: A set of operations can be defined for the potential class and these operations apply to all instances of the class.
6. **Essential requirements:** External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model. [1]

Now we will assign Selection criteria scores to the 23 potential classes.

Potential Class	**SC Score**
Users	3,4,5,6 (Selected)
System	2,3
Admin User	1,3
Production Manager	1,3
Mechanic	1.3
Supervisor	1,3
Quality Checker	1,3
Viewer	3
Resources	1,3,4,5,6 (Selected)

Visual Sensor	1,3, 6 (Selected)
Assembly Line	1,2,3,4,5 (Selected)
Product	3,4,5 (Selected)
Assembly Line Layout	1,2,3,4,5 (Selected)
Matrices	2,3,4,5 (Selected)
Line Chief	1,3
Production	1,3,4,5,6 (Selected)
Guest User	2,3
Issues	3,4,5 (Selected)
Congestion	3,4,5,6 (Selected)
Request	2,3,4,5
Report	1,2,3,6
Line Balancing	3,4,5 (Selected)
Quality	1,3,5

Selected Classes are:

1. Users
2. Resources
3. Visual Sensor
4. Assembly Line
5. Product
6. Assembly Line Layout
7. Matrices
8. Production
9. Issues
10. Congestion
11. Line Balancing
12. Request
13. Report
14. Quality

5.2 Verb List

1. adds users
2. Edit information
3. inserts resources
4. inserts the number of assembly line
5. inserts the design file
6. inserts the number of products to be produced
7. selects assembly line
8. sets the total number of product
9. defines the assembly line layout
10. using an interactive layout interface
11. generates an assembly line layout
12. based on available information
13. change layout
14. shows useful matrices
15. confirm layout
16. Send notification
17. define tolerance level of error
18. Set daily target
19. assign line chief
20. view assembly line layout
21. set hourly targets
22. adds supervisors
23. approved assembly line layouts
24. upload tutorials
25. View tutorial
26. View line
27. reaches limit
28. identify congestion
29. Updated layout
30. Mark congested workstation
31. insert congestion area
32. check reason
33. Notified
34. request mechanic
35. Accept request
36. reject request
37. fixing the machine
38. inserts failure issue
39. request workstation

40. Accept resource allocation
41. Rejects resource allocation
42. validate request
43. forward request
44. Review request
45. Detect over workstation
46. Detect under workstation
47. mark workstation
48. performs line balancing
49. check workstation
50. Fixing workstation information
51. generates machine reports
52. check products
53. input number of accepted product
54. Input number of rejected products
55. validate hourly production count
56. Store backpropagation information
57. Exceed tolerance level
58. Inspect
59. insert the issue of the quality drop
60. reviews the overall production
61. Broadcast line
62. Print report
63. Archive report

5.3 List of Objects

Users

Attributes: Admin user, Production Manager, Mechanic, Supervisor, Quality Checker, Notification, Viewer, Line Chief, Guest User
Methods: add_user(), edit_user_Info(), set_default_password(), change_password(), logs_in(), notify_mechanic(), notify_user(), notify_LC(), set_viewer_info(), authenticate(), set_username(), set_title(), set_general_description()

Resources

Attributes: Overlock sewing machine, Assembly Table, Marking Table, Electronic Sewing Machine, Types of Machine, Number of Machines, Workstation, Available Resources, Workstation Fixing Information, Allocated Workstation, Number of Broken machines, Number of Fixed machines, Machine status
Methods: insert_total_resources(), inerts_number_of_assembly_line(), insert_general_info(), insert_machine_failure_reason(),check_workstation(), record_machine_issue()

Visual Sensor

Attributes: Stack Height, Limit,
Methods: monitor_stacks_in_assembly_line(), stack_height_reaches_limit()

Assembly Line

Attributes: Number of Assembly Lines, Available Information,
Methods: select_assembly_line(), assign_line_chief(), available_information()

Quality

Attributes: Number of Accepted Products, Number of Rejected Products, Issue of Quality Drop, Quality Index, Quality Issues per Workstation, Hourly Rejected Product, (Causes of Quality Fall)
Methods: get_tolerance_level(), input_accepted_product_count(), input_rejected_product_count(), quality_exceed_tolerance_level(), generate_alert_notification(), calculate_quality_drop(), record_quality_drop_reason()

Product

Attributes: Design File, Types of Tasks, Sizes of the Cutting Pieces, Size of the Final Product, Tolerance Level of Error, Piece of Product, Tutorials,
Methods: insert_design_file(), define_tolerance_level(), upload_tutorial(), view_tutorial()

Assembly Line Layout

Attributes: Live Layout,
Methods: , define_assembly_line_layout(), generate_assembly_line_layout(), change_layout() confirm_layout(), notifyLayoutChange(), view_layout(),

Matrices

Attributes: Line efficiency, Resource Utilization Index, Possible Bottleneck points, Assembly Line Performance Matrices, Machine Issues Index
Methods: : show_line_efficiency(), show_resource_utilization_index(), Possible Bottleneck points(),

Production

Attributes: Number of Products, Targets, Hourly Targets, Counting, Total number of Productions vs Target, Required Overtime, Number of Production Deficit, Target per Workstation, Hourly Completed Product, Average Cycle Time, Target Production, Reached Production,
Methods: inserts_number_of_products(), sets_number_of_product_to_produce(), notify_production_target(), set_hourly_target(), get_product_count(), validate_hourly_production_report(), overtime_calculation(), notify(), start_production(), end_production()

Issues

Attributes: Reason for Congestion, Worker ’s Learning Issues, New Task, Problem, Machine Failure Issue, Over-workstation Issues, Issues that happened throughout the day, Fall of Production, Causes of Quality Fall

Methods: instruct_supervisor_due_to_congestion(), check_reason_for_congestion(), record_reason_of_congestion(), notify_line_chief(), request_mechanic(), notify_unavailability() get_quality_issues(), get_machine_issues()

Congestion

Attributes: Possible Bottleneck points, Assembly Line Stacks, Congested Workstation, Live Congestion Results, Congestion count,
Methods: identify_congestion(), mark_over_congestion(), mark_under_congestion(), notify()

Line Balancing

Attributes: Line Balancing Problems, Backpropagation Information, Dynamic Line Balancing,
Methods: generate_line_balancing(), insert_backpropagation_info(), backpropagate_product(), get_layout_data()

Request

Attributes: New Workstation, Requests Workstations, Approval, Allocation of Resources,
Methods: request_resource(), accept_request(), reject_request(),

Report

Attributes: Day, Machine Reports, Hourly Production Report, Production report, Hourly Production Rate, Target vs Achieved production, Daily Production Report, Production per hour of the day, Quality Assurance Report, Assembly line graph of quality drop, Machine Report, Time Wasted, Congestion Report, Congestion graph of the assembly line layout, Efficiency vs Congestion graph, Line Balancing Report, Bottleneck points graph, Time Series Graph, Monthly Assembly line performance report
Methods:generate_machine_report(), generate_hourly_production_report(), generate_quality_report(), generate_daily_production_report(), generate_line_balancing_report(), generate_congestion_report(), generate_graph(), get_calculated_matrices(), notify(), download_report()

5.4 Class Analysis

Our raw classes have many uncanny methods and attributes. Hence we polish the classes in this stage.

Users

Attributes: Admin user, Production Manager, Mechanic, Supervisor, Quality Checker, Notification, Viewer, Line Chief, Guest User
Methods: add_user(), edit_user_Info(), set_default_password(), change_password(), logs_in(), notify_mechanic(), notify_user(), notify_LC(), set_viewer_info(), authenticate(), set_username(), set_title(), set_general_description()

Resources

Analysis: “Overlock sewing machine, Assembly Table, Marking Table, Electronic Sewing Machine” are just types of machine. So we will push them out of the attribute list.

Attributes: Types of Machine, Number of Machines, Workstation, Available Resources, Workstation Fixing Information, Allocated Workstation, Number of Broken machines, Number of Fixed machines, Machine status
Methods: insert_total_resources(), inerts_number_of_assembly_line(), insert_general_info(), insert_machine_failure_reason(),check_workstation(), record_machine_issue()

Visual Sensor

Attributes: Stack Height, Limit,
Methods:monitor_stacks_in_assembly_line(), stack_height_reaches_limit()

Assembly Line

Attributes: Number of Assembly Lines, Available Information,
Methods: select_assembly_line(), assign_line_chief(), available_information()

Quality

Attributes: Number of Accepted Products, Number of Rejected Products, Issue of Quality Drop,Quality Index, Quality Issues per Workstation, Hourly Rejected Product, (Causes of Quality Fall)
Methods: get_tolerance_level(), input_accepted_product_count(), input_rejected_product_count(), quality_exceed_tolerance_level(), generate_alert_notification(), calculate_quality_drop(), record_quality_drop_reason()

Product

Attributes: Design File, Types of Tasks, Sizes of the Cutting Pieces, Size of the Final Product, Tolerance Level of Error, Piece of Product, Tutorials
Methods: insert_design_file(), define_tolerance_level(), upload_tutorial(), view_tutorial()

Assembly Line Layout

Attributes: Live Layout,
Methods: , define_assembly_line_layout(), generate_assembly_line_layout(), change_layout() confirm_layout(), notifyLayoutChange(), view_layout(),

Matrices

Attributes: Line efficiency, Resource Utilization Index, Possible Bottleneck points, Assembly Line Performance Matrices, Machine Issues Index
Methods: : show_line_efficiency(), show_resource_utilization_index(), Possible Bottleneck points(),

Production

Analysis: “Total number of Productions vs Target” is implicit. “Target per Workstation, Target Production” falls inside the target array.

Attributes: Number of Products, Targets, Hourly Targets, Counting, Required Overtime, Number of Production Deficit,, Hourly Completed Product, Average Cycle Time, Reached Production,
Methods: inserts_number_of_products(), sets_number_of_product_to_produce(), notify_production_target(), set_hourly_target(), get_product_count(),

validate_hourly_production_report(), overtime_calculation(), notify(), start_production(), end_production()

Issues

Analysis: “Worker ’s Learning Issues, New Task” are congestion issues.
Attributes: Reason for Congestion, Problem, Machine Failure Issue, Over-workstation Issues, Issues that happened throughout the day, Fall of Production, Causes of Quality drop
Methods: instruct_supervisor_due_to_congestion(), check_reason_for_congestion(), record_reason_of_congestion(), notify_line_chief(), request_mechanic(), notify_unavailability() get_quality_issues(), get_machine_issues()

Congestion

Attributes: Possible Bottleneck points, Assembly Line Stacks, Congested Workstation, Live Congestion Results, Congestion count,
Methods: identify_congestion(), mark_over_congestion(), mark_under_congestion(), notify()

Line Balancing

Attributes: Line Balancing Problems, Backpropagation Information, Dynamic Line Balancing,
Methods: generate_line_balancing(), insert_backpropagation_info(), backpropagate_product(), get_layout_data()

Request

Attributes: New Workstation, Requests Workstations, Approval, Allocation of Resources,
Methods: request_resource(), accept_request(), reject_request(),

Report

Analysis: “Machine Reports, Production report, Hourly Production Rate, Daily Production Report, Quality Assurance Report, Machine Report, Congestion Report,Line Balancing Report, Monthly Assembly line performance report” - they are all an instance of Report class.

Attributes: Day, Hourly Production Report, Target vs Achieved production, Production per hour of the day, Assembly line graph of quality drop, Time Wasted, Congestion graph of the assembly line layout, Efficiency vs Congestion graph, Bottleneck points graph, Time Series Graph,
Methods:generate_machine_report(), generate_hourly_production_report(), generate_quality_report(), generate_daily_production_report(), generate_line_balancing_report(), generate_congestion_report(), generate_graph(), get_calculated_matrices(), notify(), download_report()

5.5 Merging Classes

We merge the following classes as they perform similar tasks and the classes itself don't have much responsibility so they might result in a lazy class.

1. Visual Sensor, Congestion
2. Request, Resource
3. Assembly Line, Assembly Line Layout
4. Line Balancing, Matrices

Users

Attributes: Admin user, Production Manager, Mechanic, Supervisor, Quality Checker, Notification, Viewer, Line Chief, Guest User
Methods: add_user(), edit_user_Info(), set_default_password(), change_password(), logs_in(), notify_mechanic(), notify_user(), notify_LC(), set_viewer_info(), authenticate(), set_username(), set_title(), set_general_description()

Resources

Attributes: Types of Machine, Number of Machines, Workstation, Available Resources, Workstation Fixing Information, Allocated Workstation, Number of Broken machines, Number of Fixed machines, Machine status, New Workstation, Requests Workstations, Approval, Allocation of Resources,
Methods: insert_total_resources(), inerts_number_of_assembly_line(), insert_general_info(), insert_machine_failure_reason(), request_resource(), accept_request(), reject_request(), check_workstation(), record_machine_issue()

Quality

Attributes: Number of Accepted Products, Number of Rejected Products, Issue of Quality Drop,Quality Index, Quality Issues per Workstation, Hourly Rejected Product, (Causes of Quality Fall)
Methods: get_tolerance_level(), input_accepted_product_count(), input_rejected_product_count(), quality_exceed_tolerance_level(), generate_alert_notification(), calculate_quality_drop(), record_quality_drop_reason()

Product

Attributes: Design File, Types of Tasks, Sizes of the Cutting Pieces, Size of the Final Product, Tolerance Level of Error, Piece of Product, Tutorials
Methods: insert_design_file(), define_tolerance_level(), upload_tutorial(), view_tutorial()

Assembly Line Layout

Attributes: Live Layout, Number of Assembly Lines, Available Information,
Methods: select_assembly_line(), define_assembly_line_layout(), generate_assembly_line_layout(), available_information(), change_layout() confirm_layout(), notifyLayoutChange(), assign_line_chief(), view_layout(),

Production

Attributes: Number of Products, Targets, Hourly Targets, Counting, Required Overtime, Number of Production Deficit,, Hourly Completed Product, Average Cycle Time, Reached Production,
Methods: inserts_number_of_products(), sets_number_of_product_to_produce(), notify_production_target(), set_hourly_target(), get_product_count(), validate_hourly_production_report(), overtime_calculation(), notify(),start_production(), end_production()

Issues

Attributes: Reason for Congestion, Problem, Machine Failure Issue, Over-workstation Issues, Issues that happened throughout the day, Fall of Production, Causes of Quality drop
Methods: instruct_supervisor_due_to_congestion(), check_reason_for_congestion(), record_reason_of_congestion(), notify_line_chief(), request_mechanic(), notify_unavailability() get_quality_issues(), get_machine_issues()

Congestion

Attributes: Possible Bottleneck points, Assembly Line Stacks, Congested Workstation, Live Congestion Results, Congestion count, Stack Height, Limit
Methods:monitor_stacks_in_assembly_line(), identify_congestion(), stack_height_reaches_limit(), mark_over_congestion(), mark_under_congestion(), notify()

Line Balancing

Attributes: Line Balancing Problems, Backpropagation Information, Dynamic Line Balancing, Line efficiency, Resource Utilization Index, Possible Bottleneck points, Assembly Line Performance Matrices, Machine Issues Index
Methods: show_line_efficiency(), show_resource_utilization_index(), Possible Bottleneck points(), generate_line_balancing(), insert_backpropagation_info(), backpropagate_product(), get_layout_data()

Report

Attributes: Day, Hourly Production Report, Target vs Achieved production, Production per hour of the day, Assembly line graph of quality drop, Time Wasted, Congestion graph of the assembly line layout, Efficiency vs Congestion graph, Bottleneck points graph, Time Series Graph,
Methods:generate_machine_report(), generate_hourly_production_report(), generate_quality_report(), generate_daily_production_report(), generate_line_balancing_report(), generate_congestion_report(), generate_graph(), get_calculated_matrices(), notify(), download_report()

5.6 Collaborator

We considered the class responsibility and story to find the list of collaborators.

Class	Collaborator
User	Resource, Product
Resource	Issue, Assembly Line Layout
Quality	Line Balancing, Issue
Product	Quality, Production
Assembly Line Layout	Line Balancing, User, Congestion
Production	Quality, Report User
Issue	Resource, Quality, Line Balancing
Congestion	User, Issue, Assembly Line Layout
Line Balancing	Assembly Line Layout, Quality
Report	Issues, Production, User, Line Balancing

5.7 Class Card

Class: User

Class: User	
Attribute	**Method**
Admin user, Production Manager, Mechanic, Supervisor, Quality Checker, Notification, Viewer, Line Chief, Guest User	add_user() edit_user_Info() set_default_password() change_password(), logs_in(), notify_mechanic(), notify_user(), notify_LC() set_viewer_info() authenticate(), set_username(), set_title(), set_general_description()
Responsibility	**Collaborator**
1. The admin can add users to the system. 2. The user can change their user information. 3. Admin can add viewer information. 4. The users can change their account password. 5. Production manager, line chief, supervisor, mechanic, quality checker, admin, viewer and guest users have their own login system. 6. Production manager is notified on hourly production report, request for resource, congestion information 7. Each user will have a different level of access to the system. 8. Notification system to notify about any new reports or messages from other users. 9. The system must have an admin account.	Resource, Product

class : Resource

Class: Resource	
Attribute	**Method**
Types of Machine, Number of Machines, Workstation, Available Resources, Workstation Fixing Information, Allocated Workstation, Number of Broken machines, Number of Fixed machines, Machine status, New Workstation, Requests Workstations, Approval, Allocation of Resources,	insert_total_resources(), inerts_number_of_assembly_line(), insert_general_info(), insert_machine_failure_reason(), request_resource(), accept_request(), reject_request(), check_workstation(), record_machine_issue()
Responsibility	**Collaborator**
1. Production managers can input the amount of resources that is available to him. 2. Line chief can request for extra resources. 3. Mechanics accept/reject workstation fixing requests and insert workstation information(fixed or not, what type of problem occurred) 4. Mechanics uploads daily workstation details	Issue, Assembly Line Layout

Class: Product

Class: Product	
Attribute	**Method**
Design File, Types of Tasks, Sizes of the Cutting Pieces, Size of the Final Product, Tolerance Level of Error, Piece of Product, Tutorials	insert_design_file(), define_tolerance_level(), upload_tutorial(), view_tutorial(),
Responsibility	**Collaborator**
1. Production manager can upload product type and the different types of tasks required to make the product as the design file 2. Production manager defines fault tolerance levels. 3. Supervisor can view product details. 4. Supervisors upload tutorials for workstations. 5. Guest users can view tutorials.	Production, Report, Quality

Class: Assembly Line Layout

Class: Assembly Line Layout	
Attribute	**Method**
Live Layout, Number of Assembly Lines, Available Information	select_assembly_line(), define_assembly_line_layout(), generate_assembly_line_layout(), available_information(), change_layout() confirm_layout(), notifyLayoutChange(), assignLineChief(), view_layout(),
Responsibility	**Collaborator**
1. Production manager can create the assembly line layout. 2. Production manager finalizes the assembly line layout. 3. Line chief can view layout and line targets 4. Supervisors can view assembly line layouts. 5. Viewers can view live assembly line status, production reports and other informations made available by the admin 6. Live layout of an assembly line must be updated and viewable at all times 7. Generate layout based on design file and resources 8. Production manager can assign the line to a line chief.	Report, User

Class: Production

Class: Production	
Attribute	**Method**
Number of Products, Targets, Hourly Targets, Counting, Required Overtime, Number of Production Deficit,, Hourly Completed Product, Average Cycle Time, Reached Production,	inserts_number_of_products(), sets_number_of_product_to_produce(), notify_production_target(), set_hourly_target(), get_product_count(), validate_hourly_production_report(), overtime_calculation(), notify()
Responsibility	**Collaborator**
1. Production manager defines: the number of products to be produced, daily line targets, 2. Line chief can set hourly line target 3. Quality checker inserts production count, and supervisor validates the count	Quality, Report, User

Class: Issue

Class: Issue	
Attribute	**Method**
Reason for Congestion, Problem, Machine Failure Issue, Over-workstation Issues, Issues that happened throughout the day, Fall of Production, Causes of Quality drop	instruct_supervisor_due_to_congestion(), check_reason_for_congestion(), record_reason_of_congestion(), notify_line_chief(), request_mechanic(), notify_unavailability() get_quality_issues(), get_machine_issues()
Responsibility	**Collaborator**
1. Line chief can request mechanics to fix workstations. 2. Supervisor uploads congestion reason 3. Quality checkers upload reasons for quality drop. 4. Mechanics upload machine issues.	Congestion, Quality, Resource

Class: Congestion

Class: Congestion	
Attribute	**Method**
Possible Bottleneck points, Assembly Line Stacks, Congested Workstation, Live Congestion Results, Congestion count, Stack Height, Limit	monitor_stacks_in_assembly_line(), identify_congestion(), stack_height_reaches_limit(), mark_over_congestion(), mark_under_congestion() notify()
Responsibility	**Collaborator**
1. Line chief will be notified of congestion information. 2. Supervisor can mark over/under workstations of the line. 3. Congestion detection in the line based on object/stack detection.	User, Issue, Assembly Line Layout

Class: Report

Class: Report	
Attribute	**Method**
Day, Hourly Production Report, Target vs Achieved production, Production per hour of the day, Assembly line graph of quality drop, Time Wasted, Congestion graph of the assembly line layout, Efficiency vs Congestion graph, Bottleneck points graph, Time Series Graph,	generate_machine_report(), generate_hourly_production_report(), generate_quality_report(), generate_daily_production_report(), generate_line_balancing_report(), generate_congestion_report(), generate_graph(), get_calculated_matrices() notify() download_report()
Responsibility	**Collaborator**
1. Line chief will be notified on hourly production reports. 2. The production manager can download reports. 3. Reports will be digitally generated automatically. 4. Communicate all the reports (production, overtime, washing, sample testing, quality assurance, machine report, congestion report, line balancing report, daily assembly line performance report) to the concerned users	Quality, Congestion, Issue, Production, Assembly Line Layout

Class: Quality

Class: Quality	
Attribute	**Method**
Number of Accepted Products, Number of Rejected Products, Issue of Quality Drop,Quality Index, Quality Issues per Workstation, Hourly Rejected Product, (Causes of Quality Fall)	get_tolerance_level(), input_accepted_product_count(), input_rejected_product_count(), quality_exceed_tolerance_level(), generate_alert_notification(), calculate_quality_drop(), record_quality_drop_reason()
Responsibility	**Collaborator**
1. Line chief will be notified on quality issues.	Product, Issue

Class: Line Balancing

Class: Line Balancing	
Attribute:	**Method**:
Line Balancing Problems, Backpropagation Information, Dynamic Line Balancing, Line efficiency, Resource Utilization Index, Possible Bottleneck points, Assembly Line Performance Matrices, Machine Issues Index	show_line_efficiency(), show_resource_utilization_index(), Possible Bottleneck points(), generate_line_balancing(), insert_backpropagation_info(), backpropagate_product(), get_layout_data()
Responsibility:	**Collaborator**
1. Supervisor uploads backpropagation information. 2. Production manager can generate performance matrices on assembly line layout 3. Automated Line balancing based on product 4. Line chief will be notified on hourly production report,	Quality, Congestion, Issue, Production, Assembly Line Layout

5.8 Class Relationship Collaboration(CRC) Diagram

CRC Diagram ID: 01 (Class: Resource)

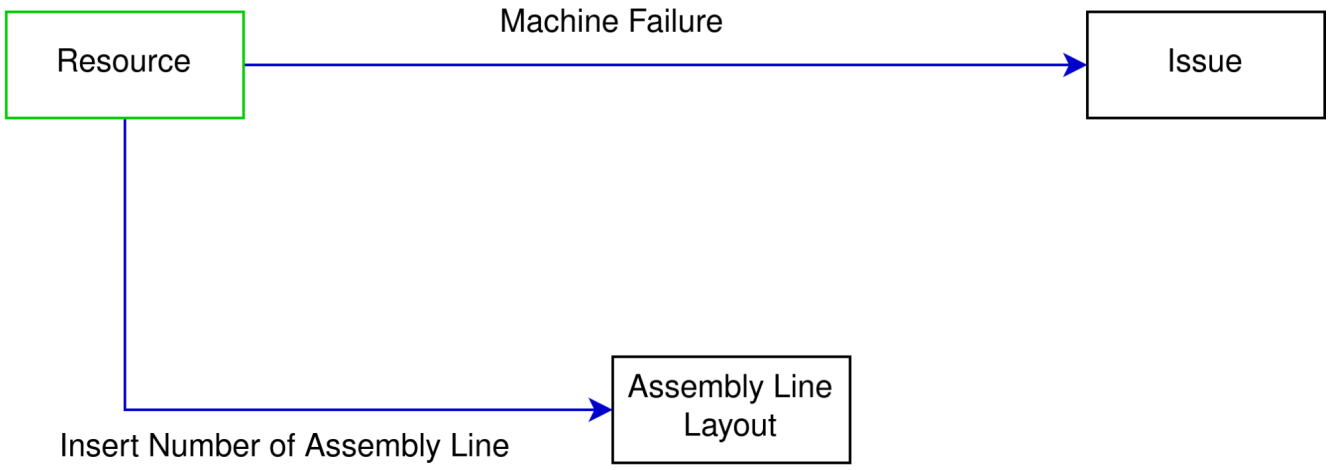


Figure 33: CRC diagram of class: resource

CRC Diagram ID: 02 (Class: Product)

Figure 34: CRC diagram of class: product

CRC Diagram ID: 03 (Class: Assembly Line Layout)

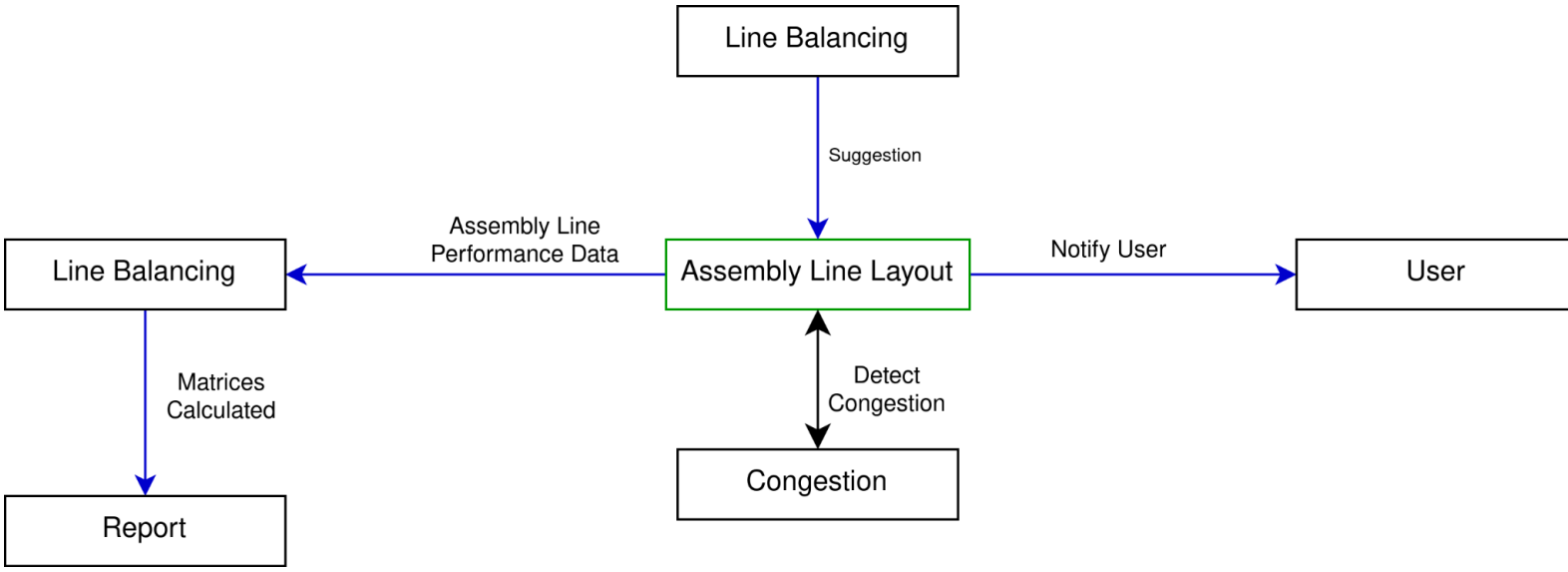


Figure 35: CRC diagram of class: Assembly Line Layout

CRC Diagram ID: 04 (Class: Production)

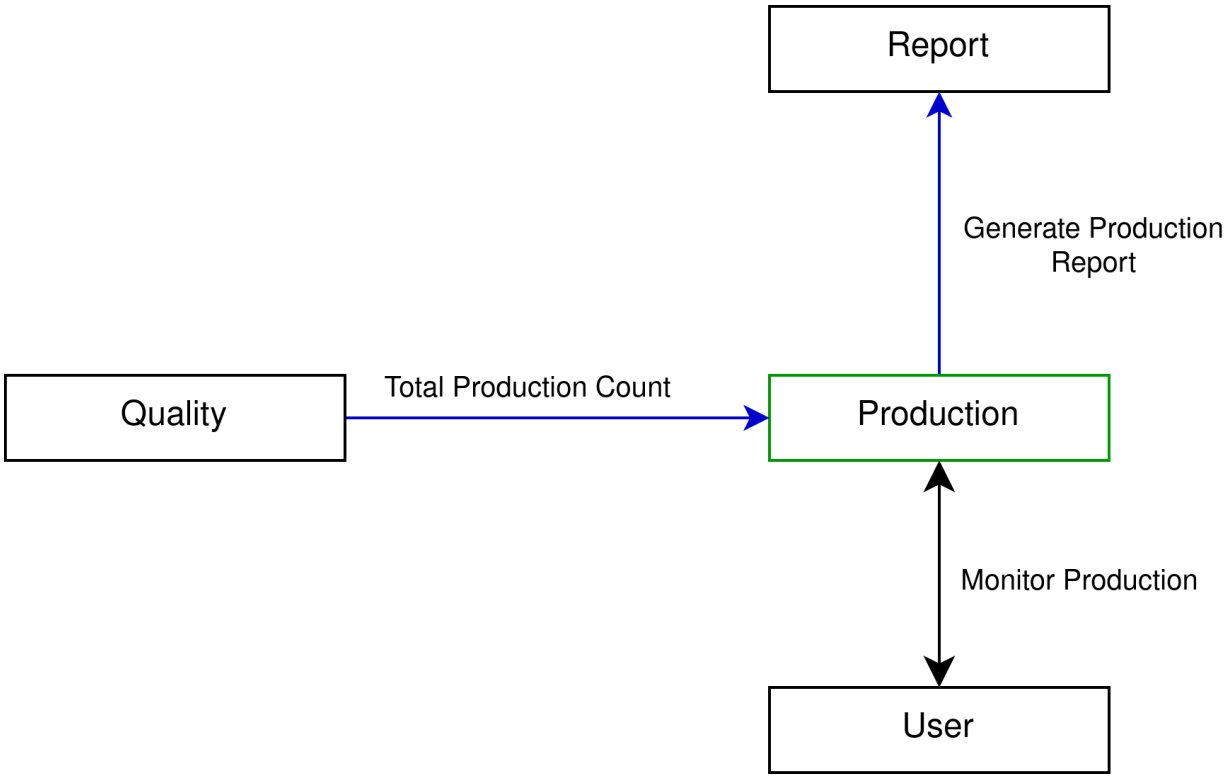


Figure 36: CRC diagram of class: Production

CRC Diagram ID: 05 (Class: Issue)

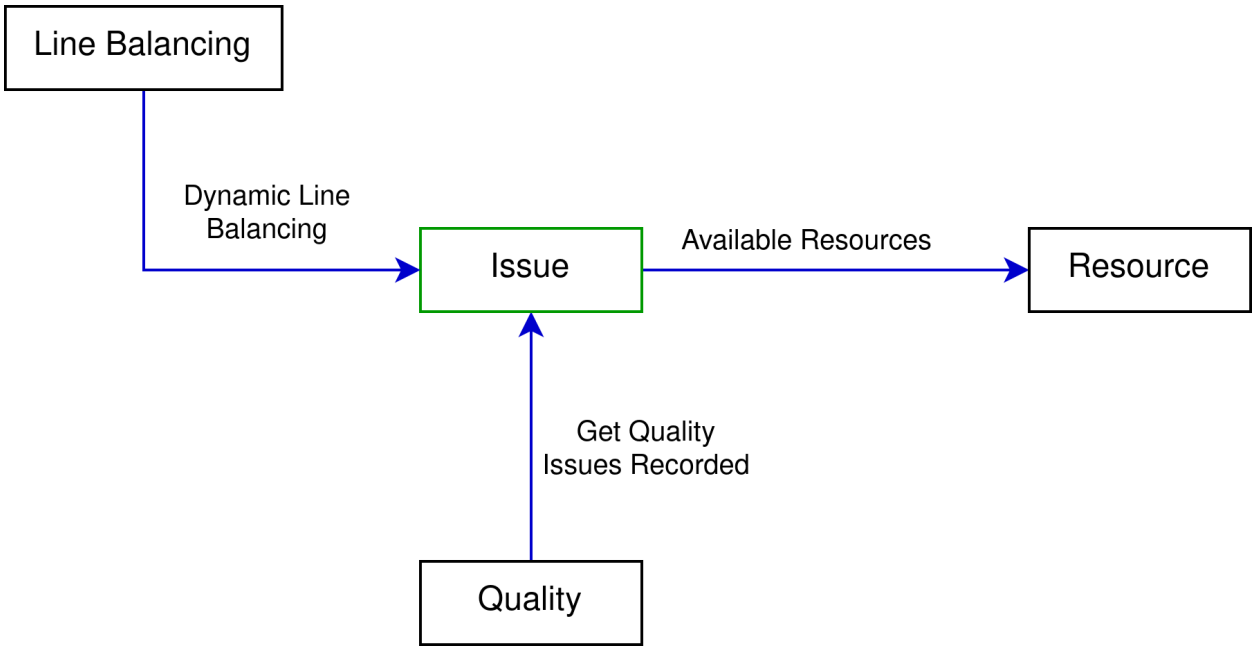


Figure 37: CRC diagram of class: Issue

CRC Diagram ID: 06 (Class: Congestion)

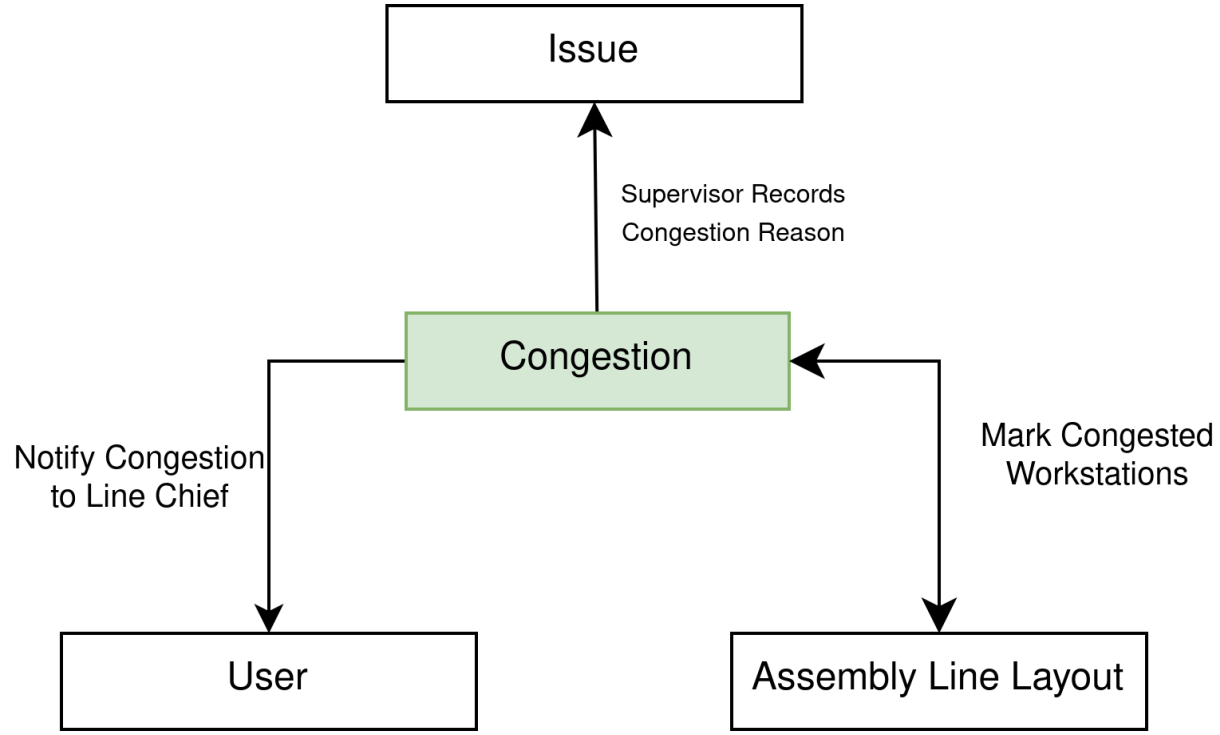


Figure 38: CRC diagram of class: Congestion

CRC Diagram ID: 07 (Class: Report)

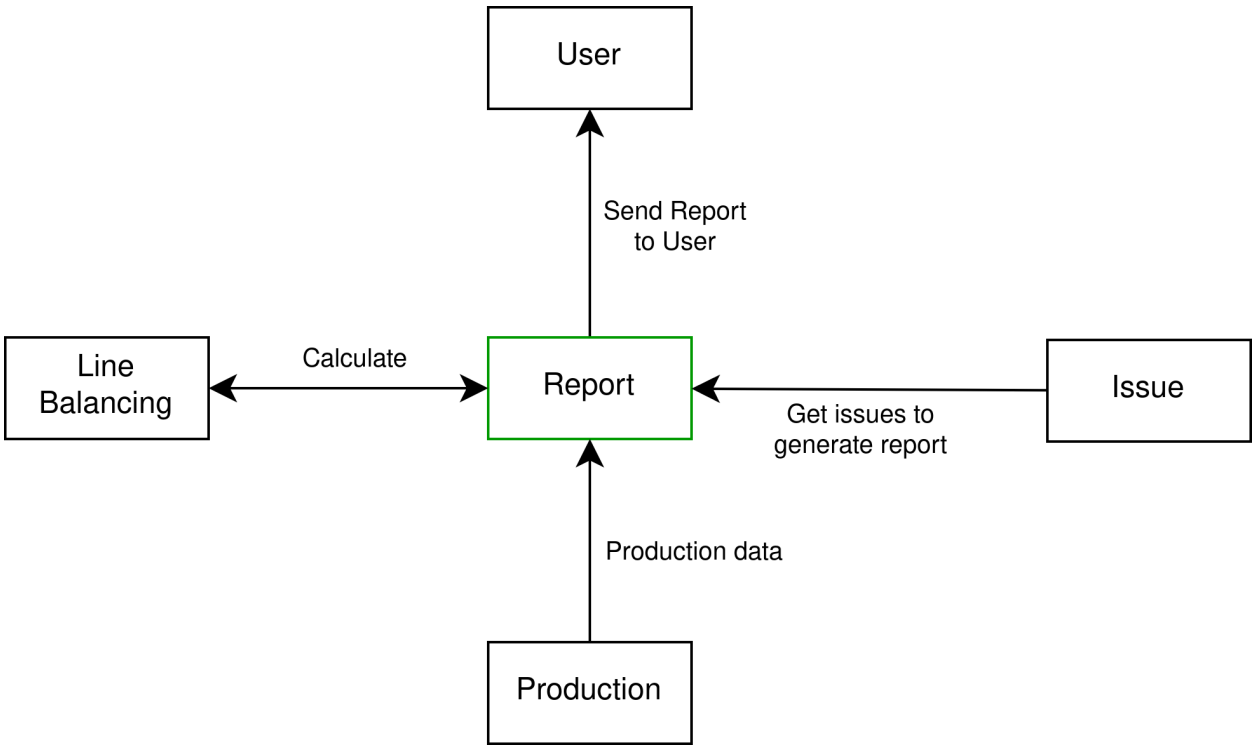


Figure 39: CRC diagram of class: Report

CRC Diagram ID: 08 (Class: Quality)

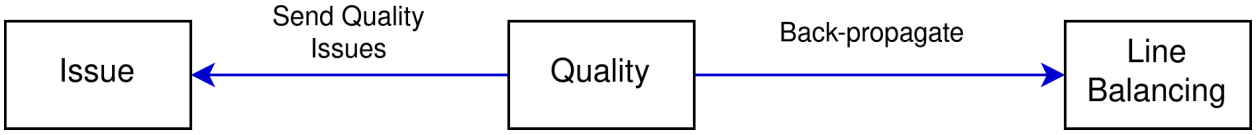


Figure 40: CRC diagram of class: Quality

CRC Diagram ID: 09 (Class: Line Balancing)

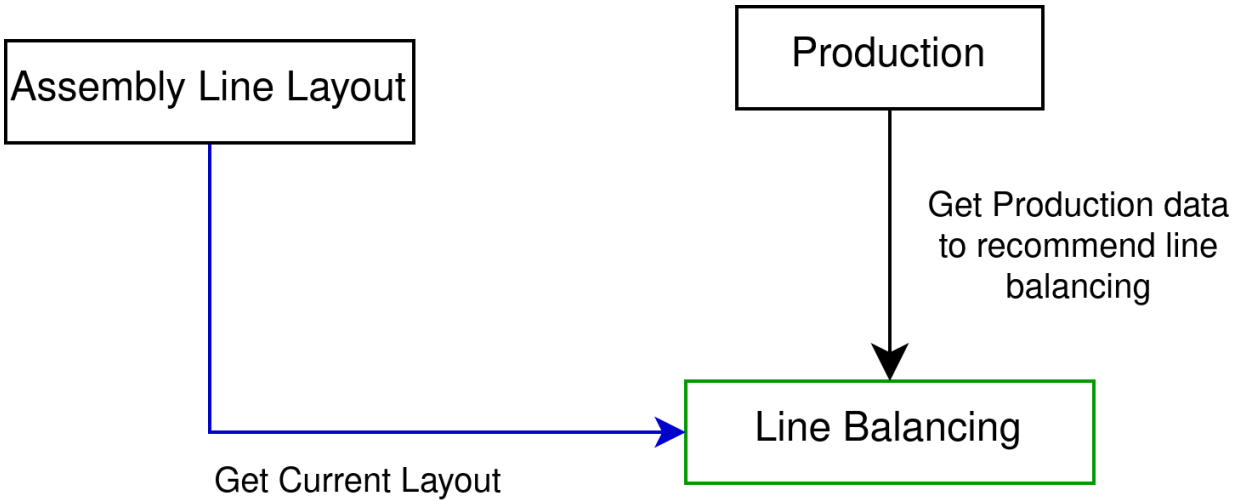


Figure 41: CRC diagram of class: Line Balancing

CRC Diagram ID: 10 (Class: User)

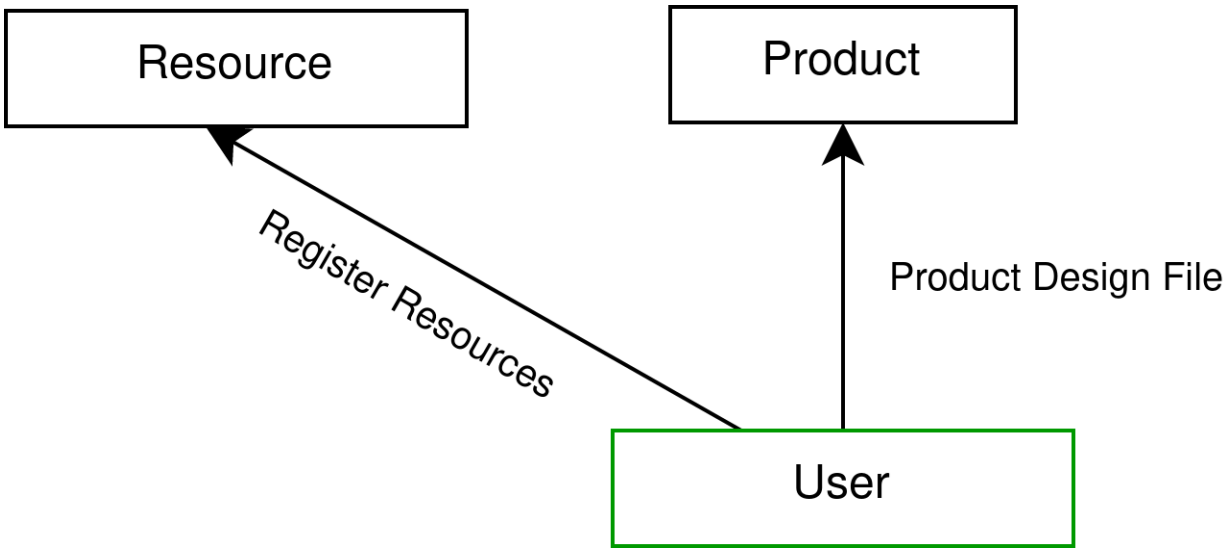


Figure 42: CRC diagram of class: User

6. Flow Oriented model

Data flow diagrams:
The data flow diagrams can be found in the following link:
https://app.diagrams.net/#G1VhcxHcZsH7SU7DHWYnjuvURMUUP9KYxT

DFD level: 0

DFD ID: 01

Name: Bird's Eye View

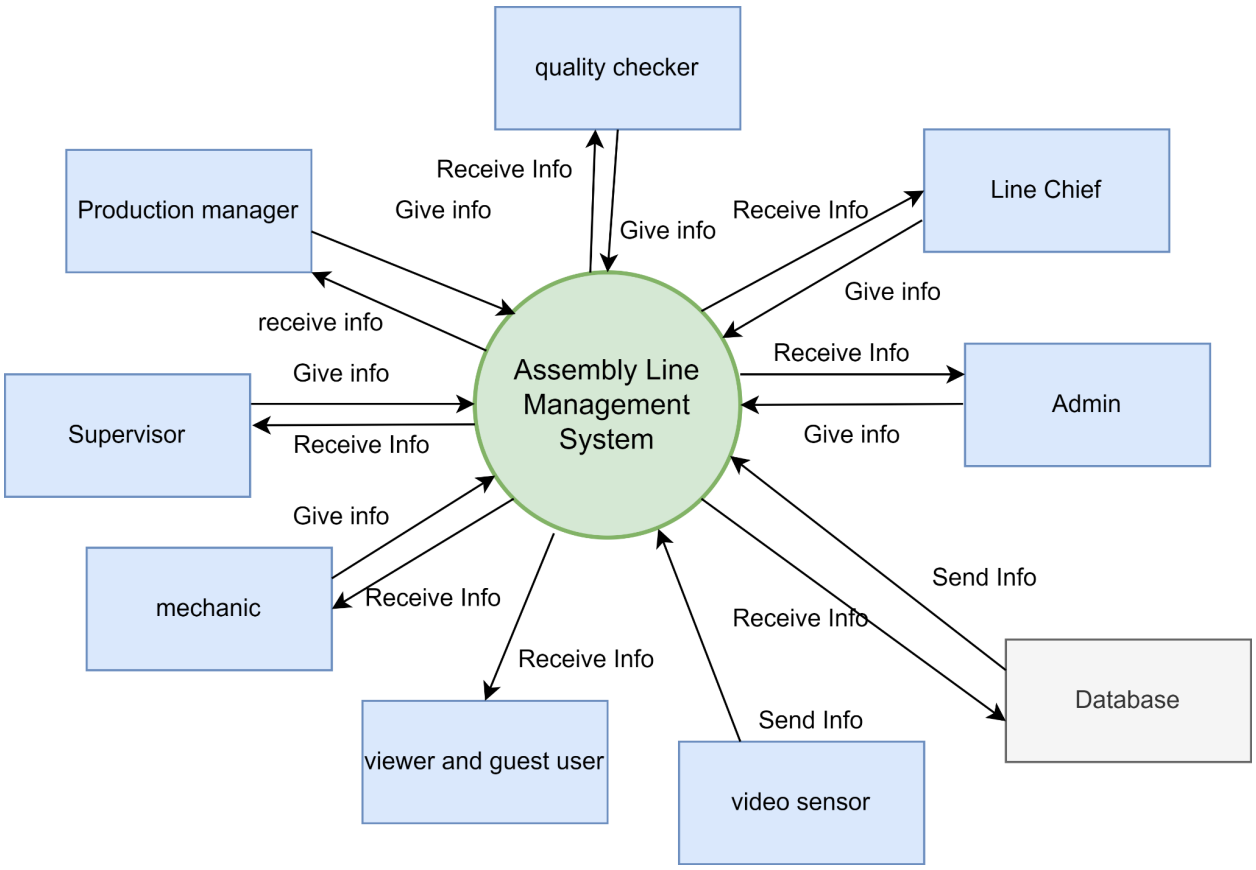


Figure 43: Level 0 DFD: Birds eye

Description: This is the level 0 DFD of the system. This will be further divided into smaller sub-systems in DFD level 1.

DFD level: 1

DFD ID: 02

Name: Assembly Line Management System

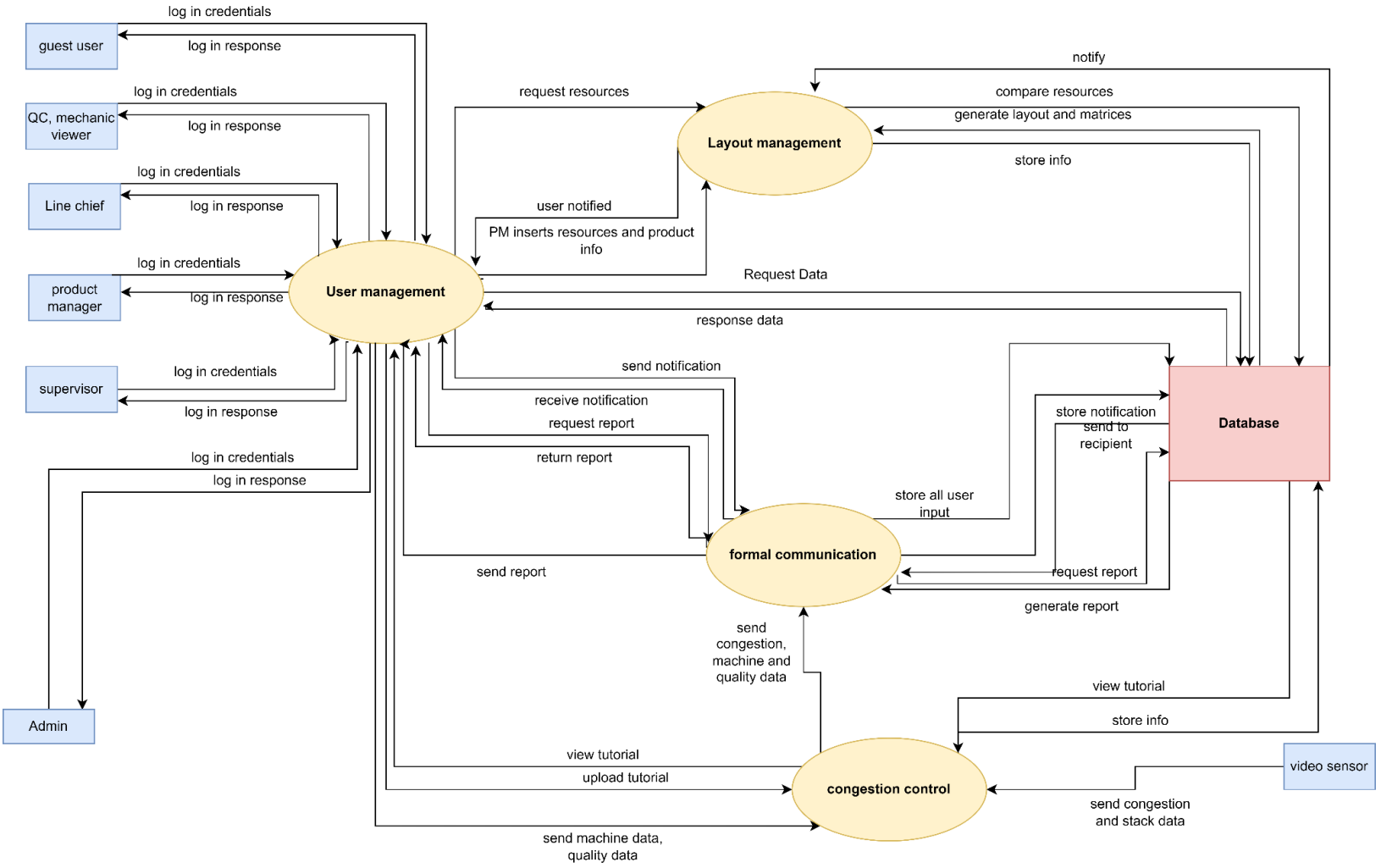


Figure 44: Level 1 DFD: Assembly Line Management

Description: This is the level 1 DFD of the Assembly Line Management System. We can see here the major subsystems of it & how data flows in between them. To make the data flow more reasonable, the major subsystems will be divided into smaller processes from level 02.

DFD level: 2

DFD ID: 03

Name: User Management

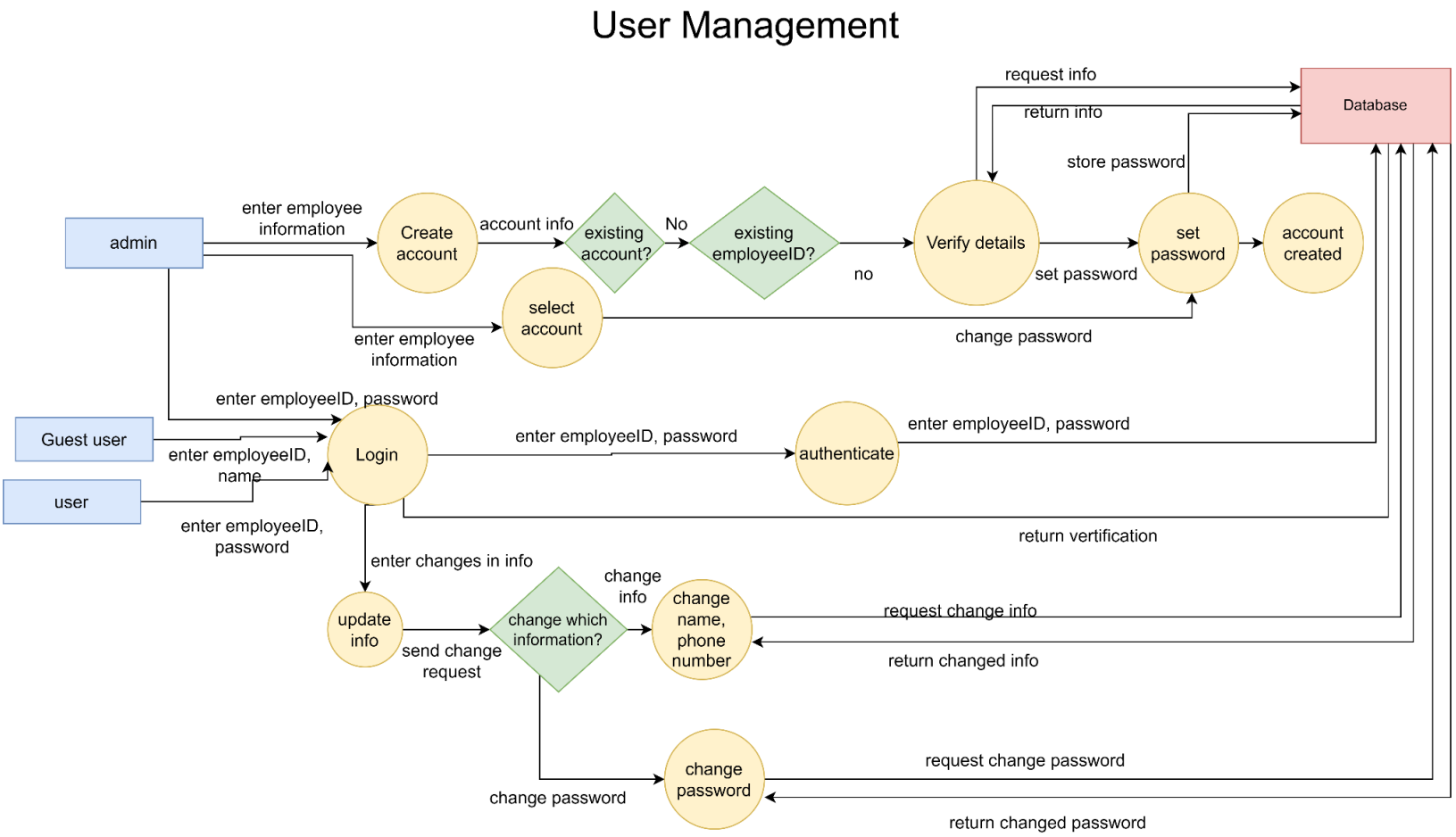


Figure 45: Level 2 DFD: User Management

Description: This is the level 2 DFD of the User Management sub-system. We can see here what and how data flows through sub-systems and processes in this level of User Management sub-system. Here employees can login and change information. Admin can create accounts and change passwords if necessary

DFD level: 2

DFD ID: 04

Name: Layout Management

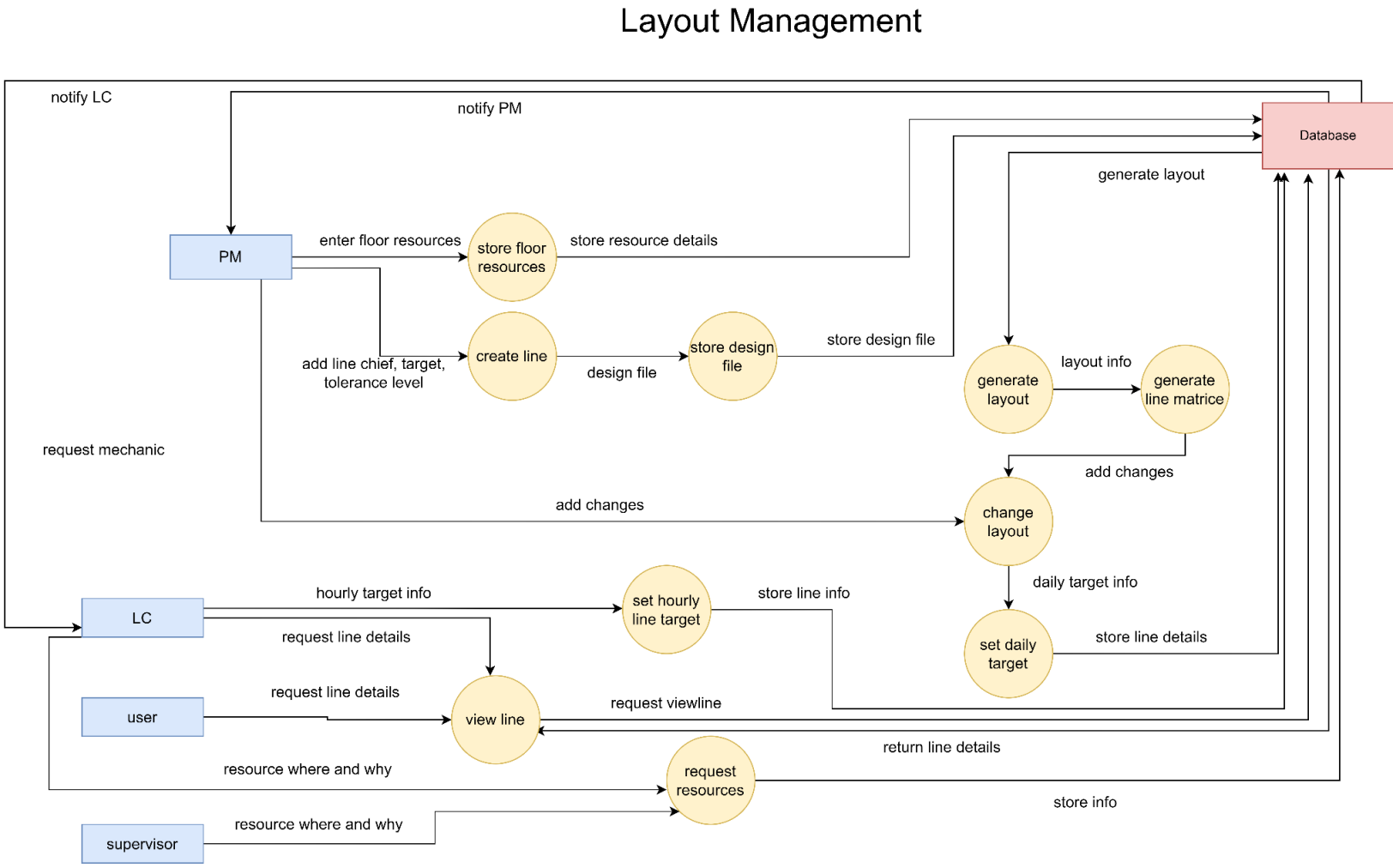


Figure 46: Level 2 DFD: Layout Management

Description: This is the level 2 DFD of the Layout Management sub-system. We can see here what and how data flows through sub-systems and processes in this level of Layout Management sub-system. Here production managers insert resources and product details to create assembly lines and their matrices. The line is viewable to all users. Production manager also sets targets and can change assembly line layouts. Line chief can request for resources.

DFD level: 2

DFD ID: 05

Name: Formal communication

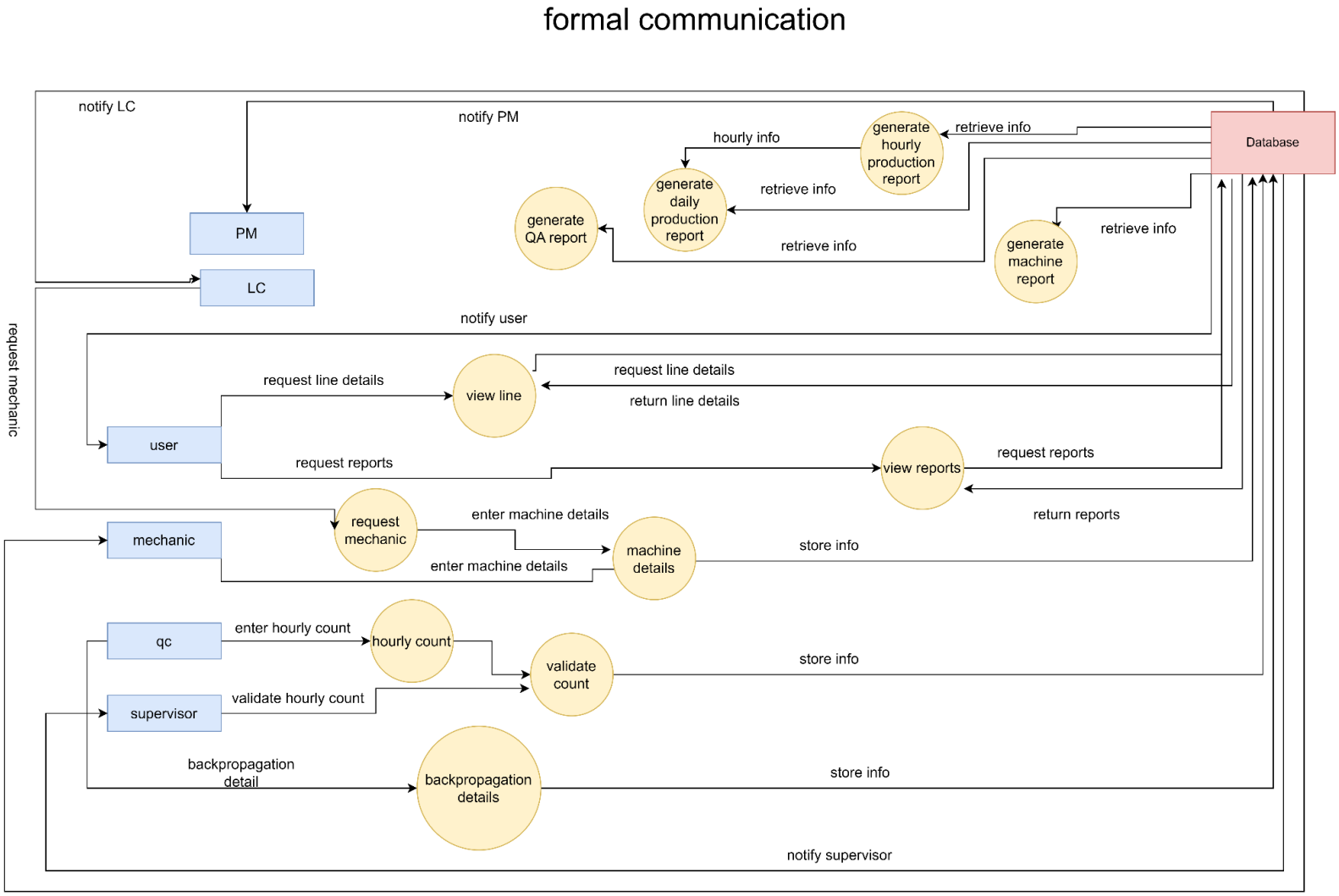


Figure 47: Level 2 DFD: Formal communication

Description: This is the level 2 DFD of the Formal Communication subsystem. We can see here what and how data flows through sub-systems and processes in this level of Formal Communication Sub-system. Here mechanic, quality assurance details are stored along with backpropagation details. And all the information generated various reports. All users can view lines and reports. The notification module is built here.

DFD level: 2

DFD ID: 06

Name: Congestion control

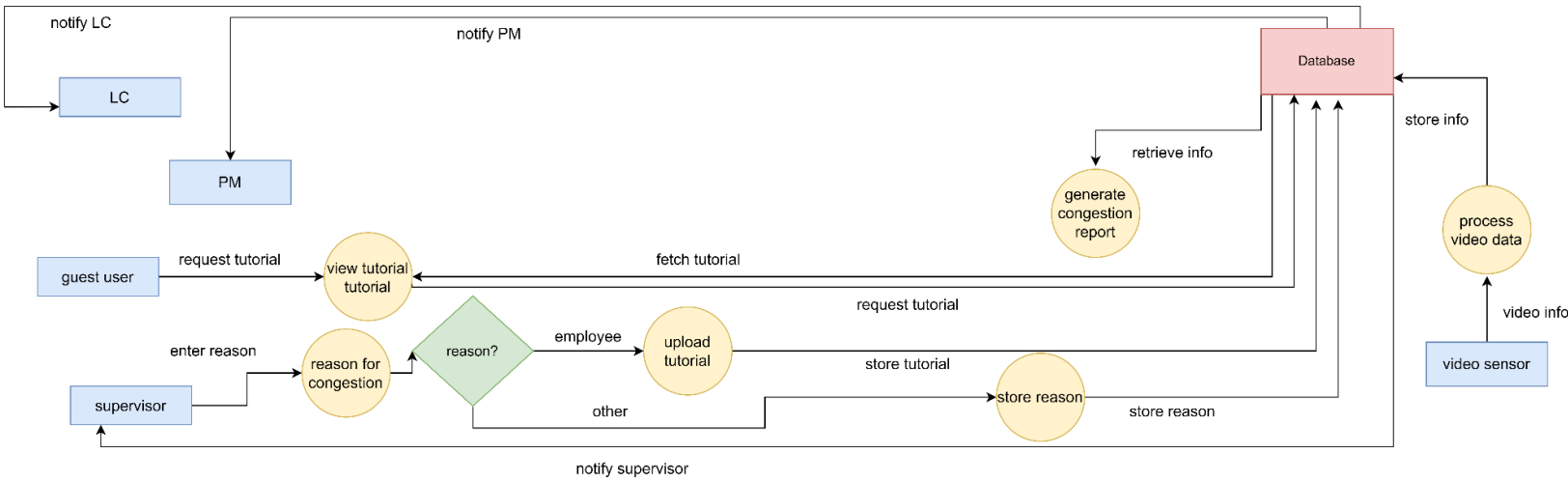


Figure 47: Level 2 DFD: Congestion control

Description: This is the level 2 DFD of the Congestion control subsystem. We can see here what and how data flows through sub-systems and processes in this level of Congestion control Sub-system. Here video sensors generate data to identify congestion and generate congestion reports. If employees are the reason, supervisors upload tutorials. For other reasons Line chief and production manager are notified to take necessary actions.

7. Behavioral Model

The Behavior Modeling indicates how the system will behave to external events or stimuli. It is represented as a function of time and event, It describes interactions between objects. It shows how individual objects collaborate to achieve the behavior of the system as a whole. In UML behavior of a system is shown with the help of use case diagram, sequence diagram and activity diagram.

To create behavioral model following things can be considered-

● Evaluation of all use-cases to fully understand the sequence of interaction within the system.
● Identification of events that drive the interaction sequence and understand how these events relate to specific classes.
● Creating sequence for each use case.
● Building a state diagram for the system.
● Reviewing the behavioral model to verify accuracy and consistency

7.1 Event List

Event ID	Event	Initiator Class	Collaborator Class
1	Add User	User	-
2	Set viewer information	User	-
3	Edit User Information	User	-
4	User Logs in	User	Production, Resource
5	Insert Resources Information	Resource	Product
6	Handling Product Design File	Product	Resource, Production
7	Communicate Report	Report	User
8	Set Production Target	Production	Product, Assembly Line Layout
9	Validate Production Status	Production	Product, Quality
10	Define assembly line layout	Assembly Line Layout	Production, Line Balancing
11	Automated Line balancing	Line Balancing	Production
12	Generate Line Matrice	Line Balancing	Quality

13	Confirm layout	Assembly Line Layout	Line Balancing
14	Notification	User, Issue	Assembly Line Layout, Congestion, Report
15	Assign	Assembly Line Layout	-
16	View assembly line layout	Assembly Line Layout	User
17	Request Resource Allocation	Resource	User
18	Validate Resource Allocation	Resource	Assembly Line Layout
19	Upload & View Tutorial	Product	Issue
20	Automatically Identify congestion	Congestion	Assembly Line Layout
21	Record congestion Reason	Issue	-
22	Request mechanic	Resource	Issue
23	Quality Inspection	Quality	-
24	Mark Over & under workstation	Congestion	Assembly Line Layout, User
25	Record Issues for Quality Drop	Quality	-
26	Handle Back Propagation	Line Balancing	Quality
27	Estimate Quality Drop	Quality	-
28	Record Machine Issue	Resource	Issue
29	Generate Report	Report	-
30	Generate Graph	Report	Issue
31	Collect Quality & Machine Issues	Issue	Production, Quality
32	Production Start-End	Production	-
33	Download reports	Report	-
34	Overtime Calculation	Production	-

7.2 State Transition Diagram

A UML state diagram is one component of a behavioral model that describes active states for each class and the events (triggers) that induce transitions between these active states.

Diagrams of the state transition diagrams can be found in the following link: https://app.diagrams.net/#G1Ad1J1-is_goKwpWl1fAfP13CS7byzHv4

STD ID: 01

Class: User

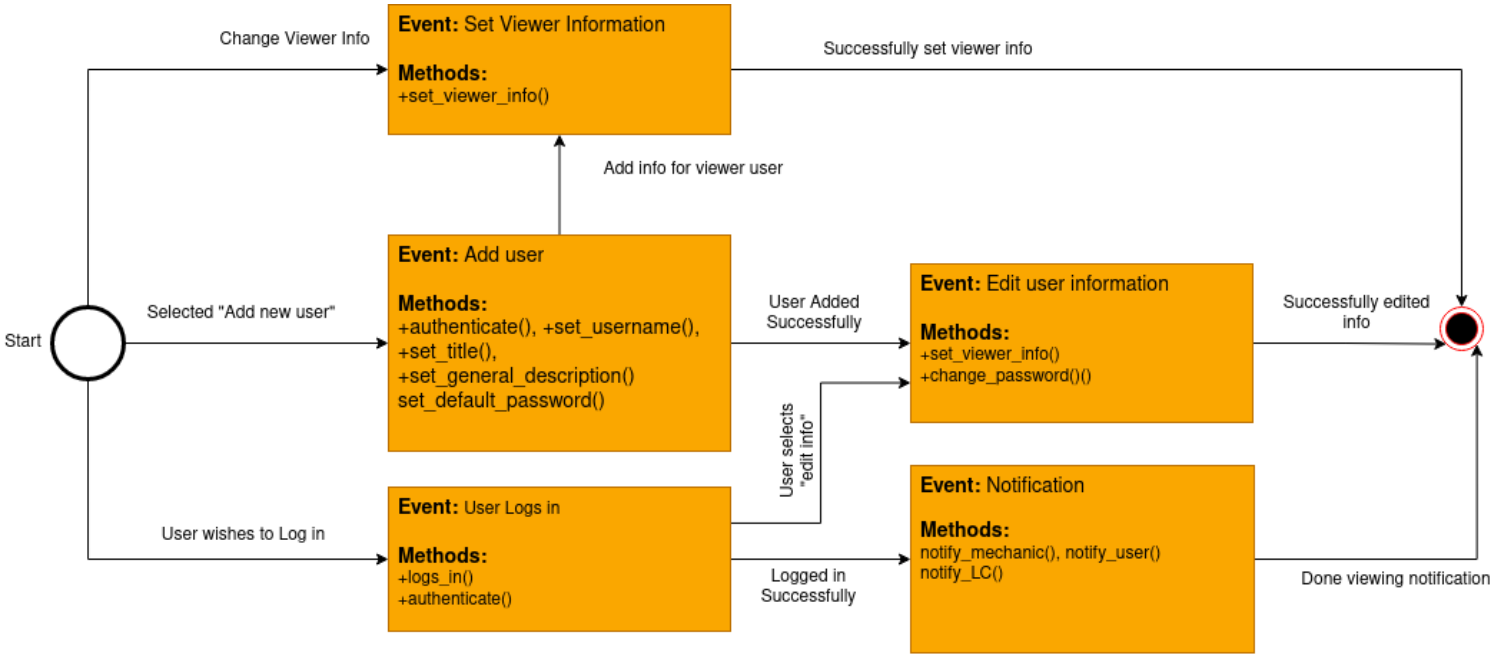


Figure 48: State transition diagram: user

STD ID: 02

Class: Assembly Line Layout

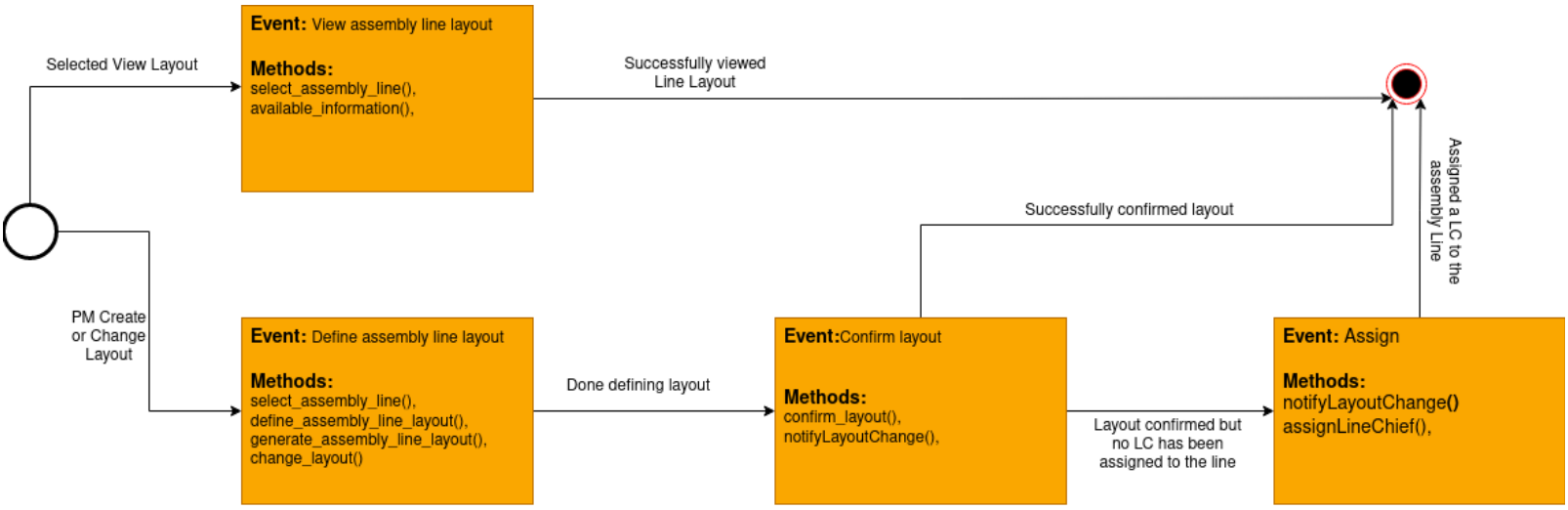


Figure 49: State transition diagram: Assembly Line Layout

STD ID: 03

Class: Line Balancing

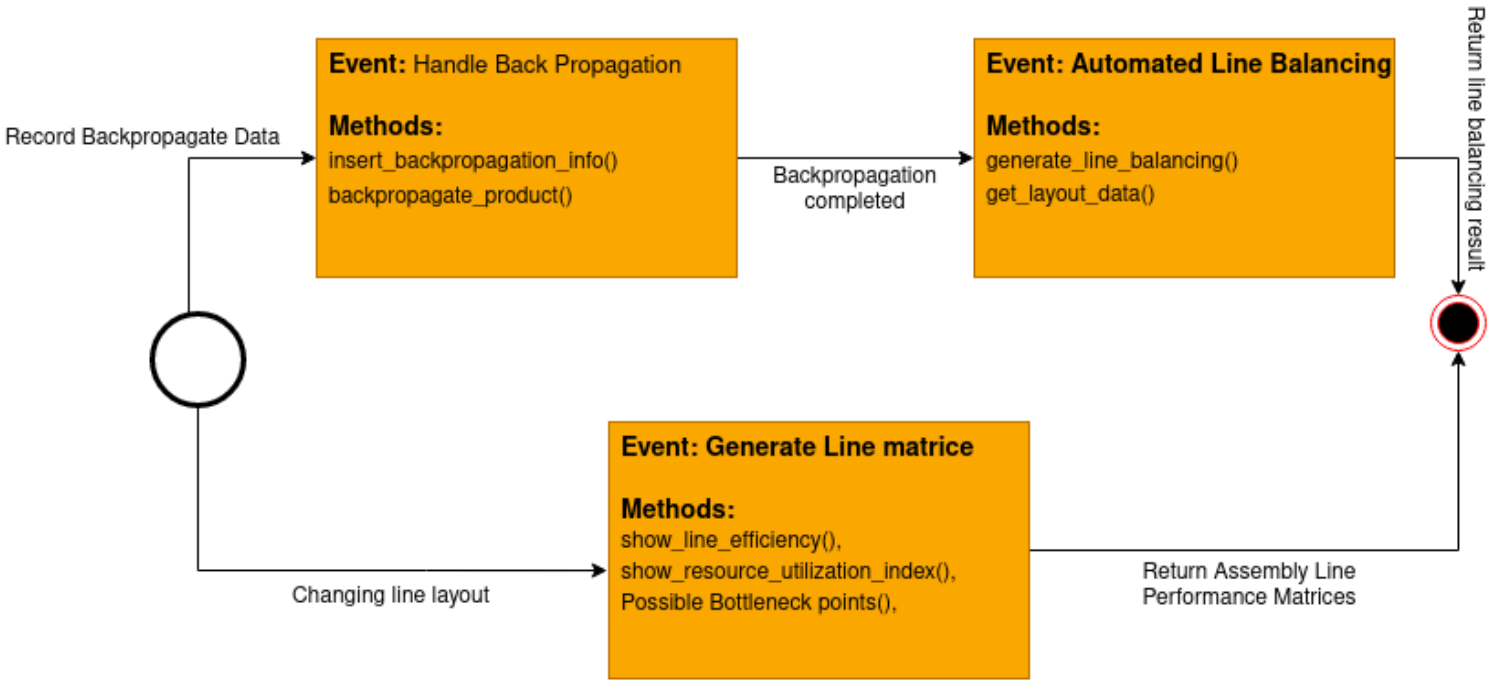


Figure 50: State transition diagram: Line Balancing

STD ID: 04

Class: Resource

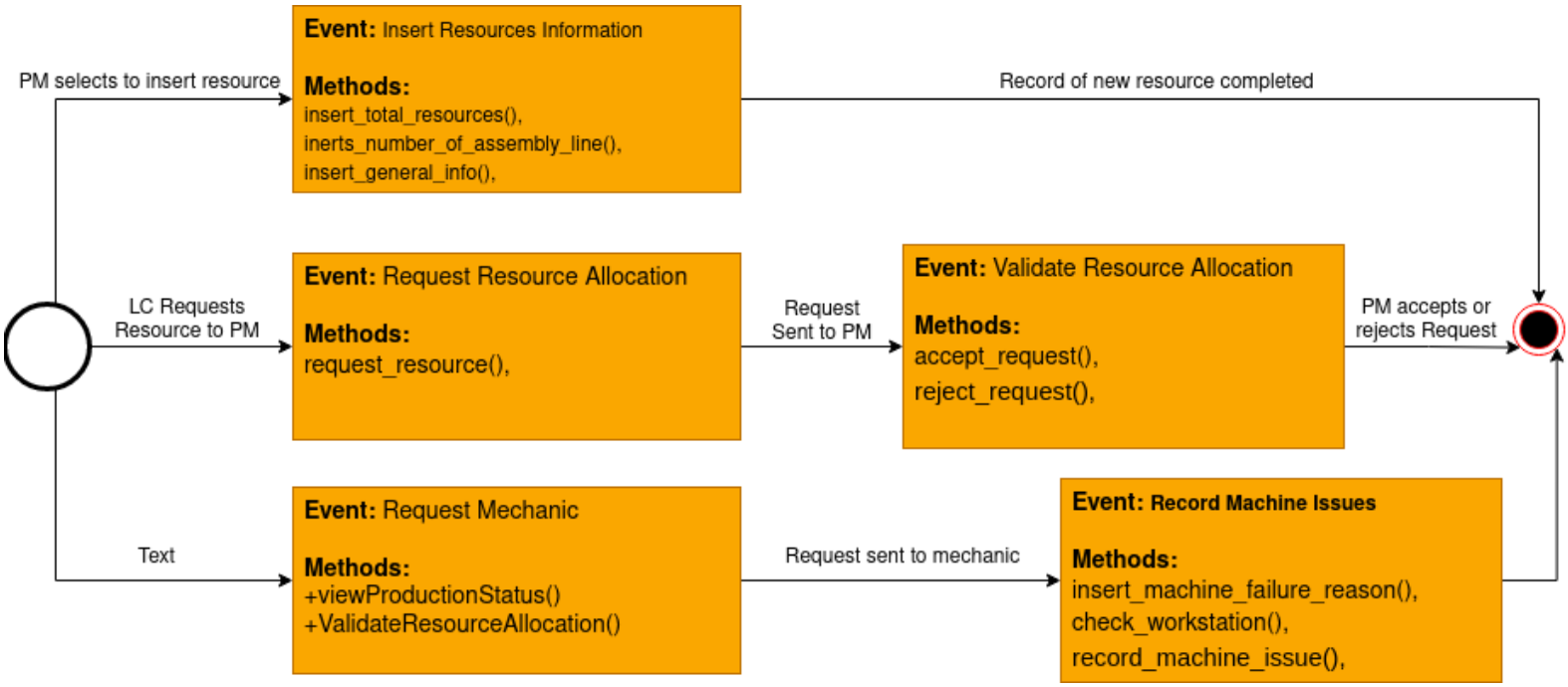


Figure 51: State transition diagram: Resource

STD ID: 05

Class: Issue

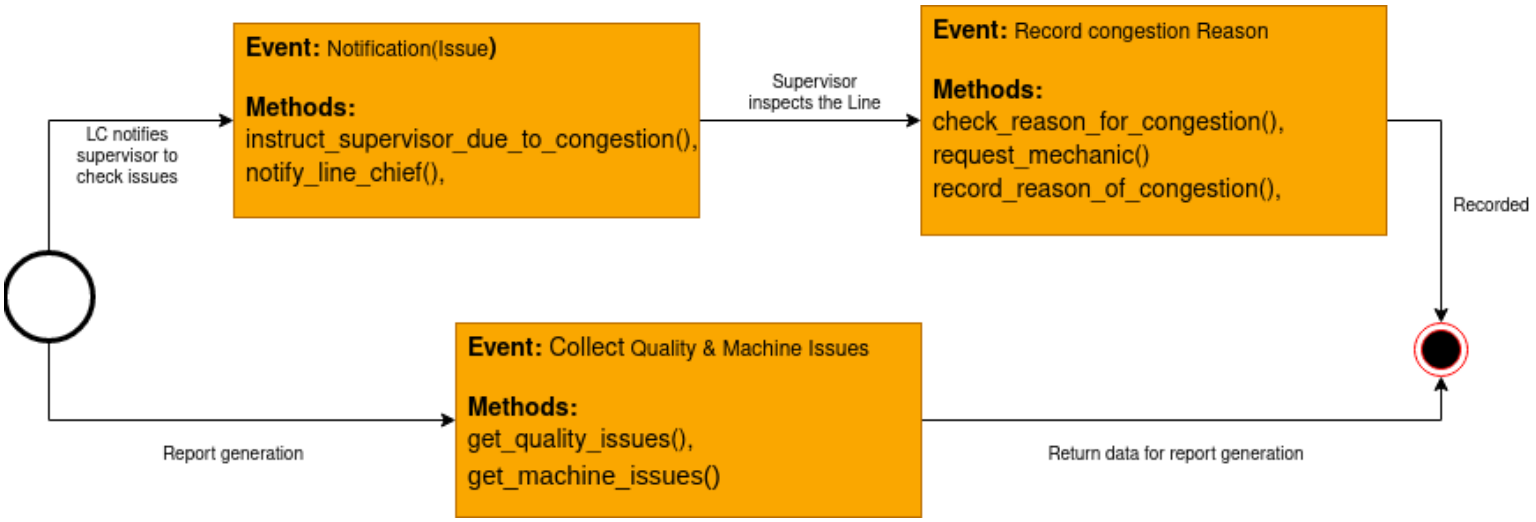


Figure 52: State transition diagram: Issue

STD ID: 06

Class: Congestion

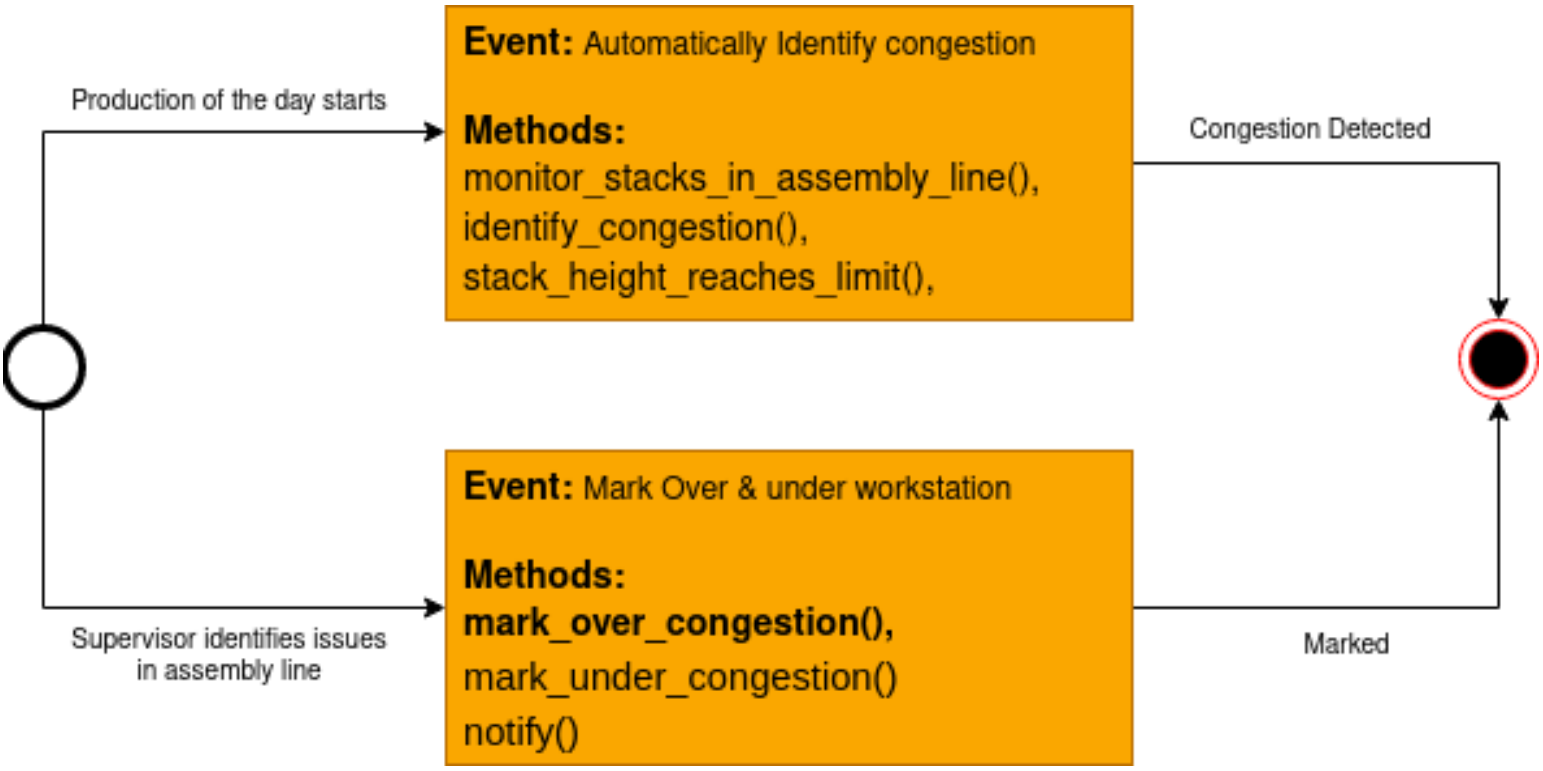


Figure 53: State transition diagram: Congestion

STD ID: 07

Class: Product

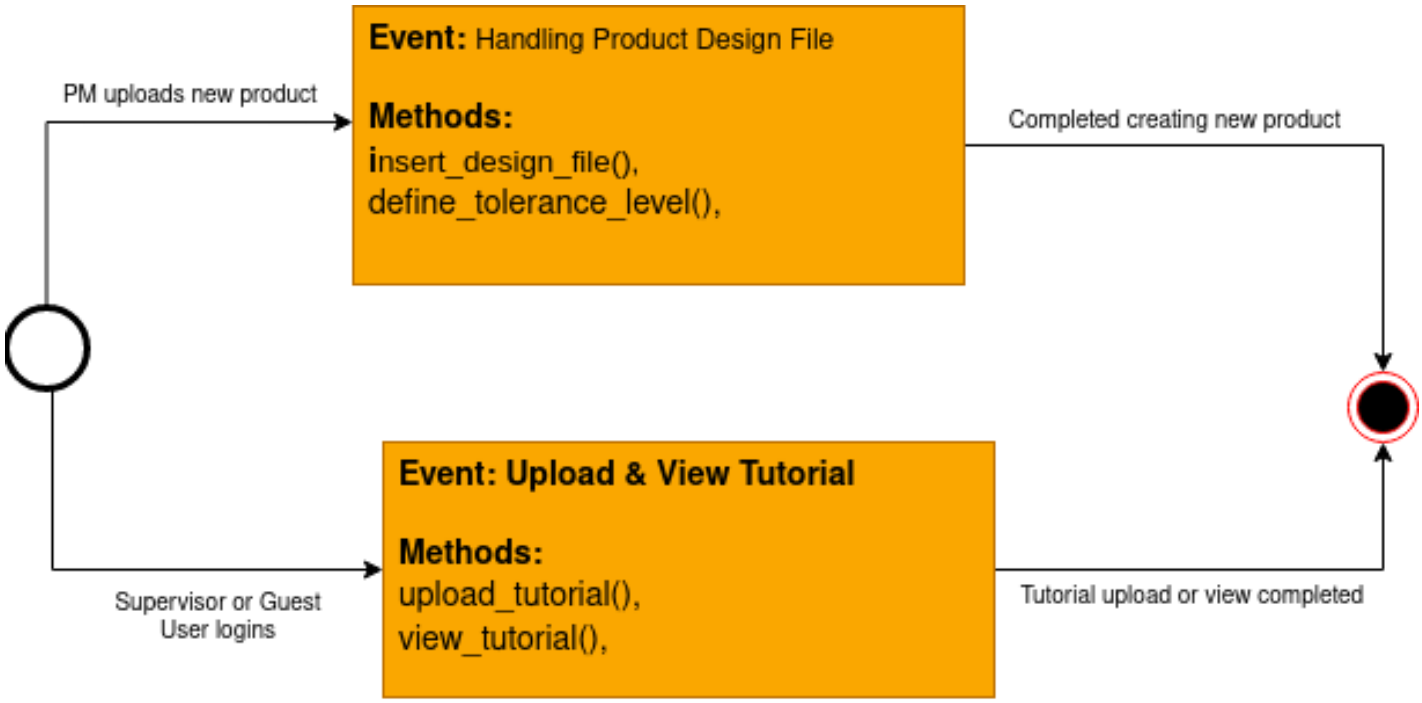


Figure 54: State transition diagram: Product

STD ID: 08

Class: Production

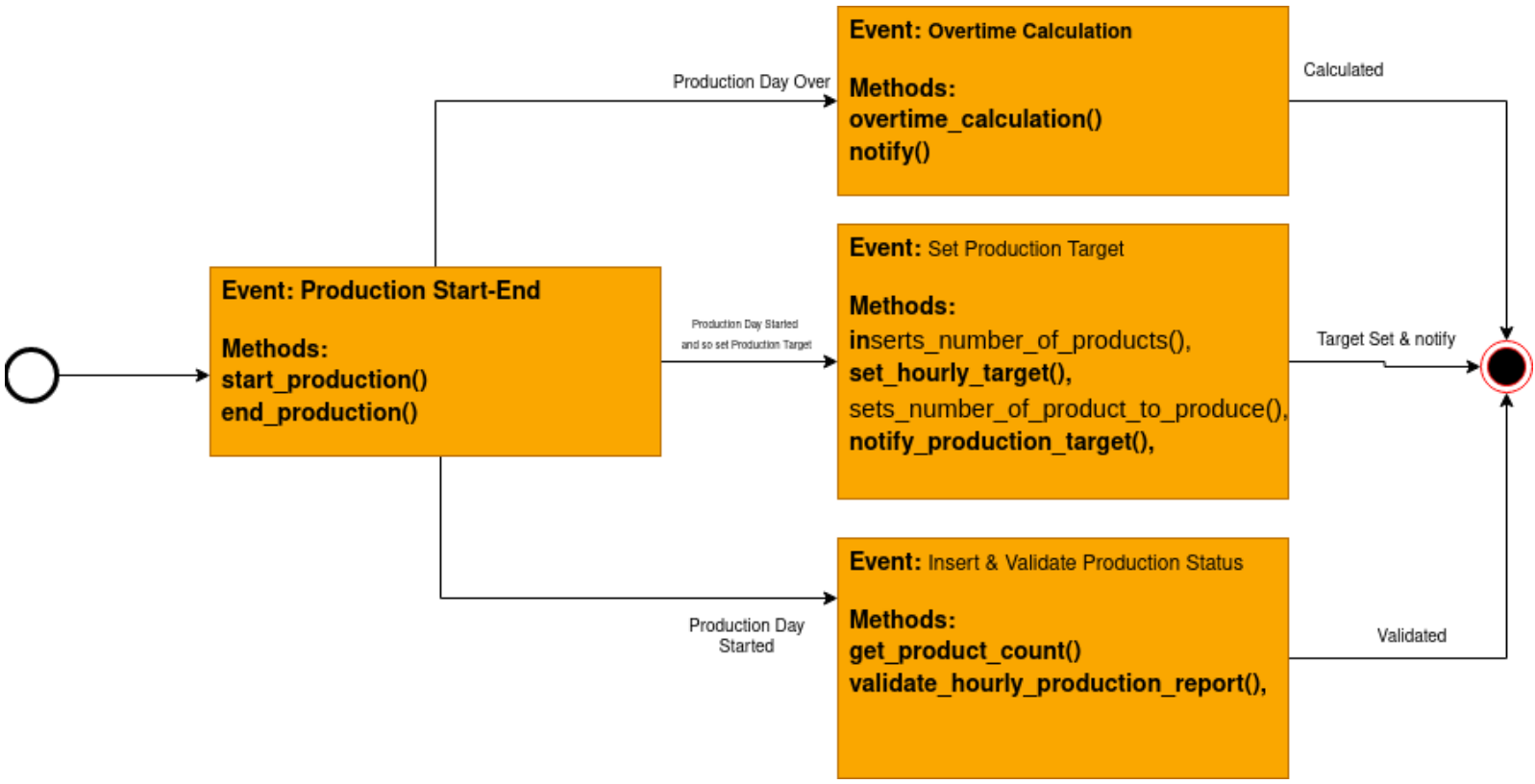


Figure 55: State transition diagram: Production

STD ID: 09

Class: Report

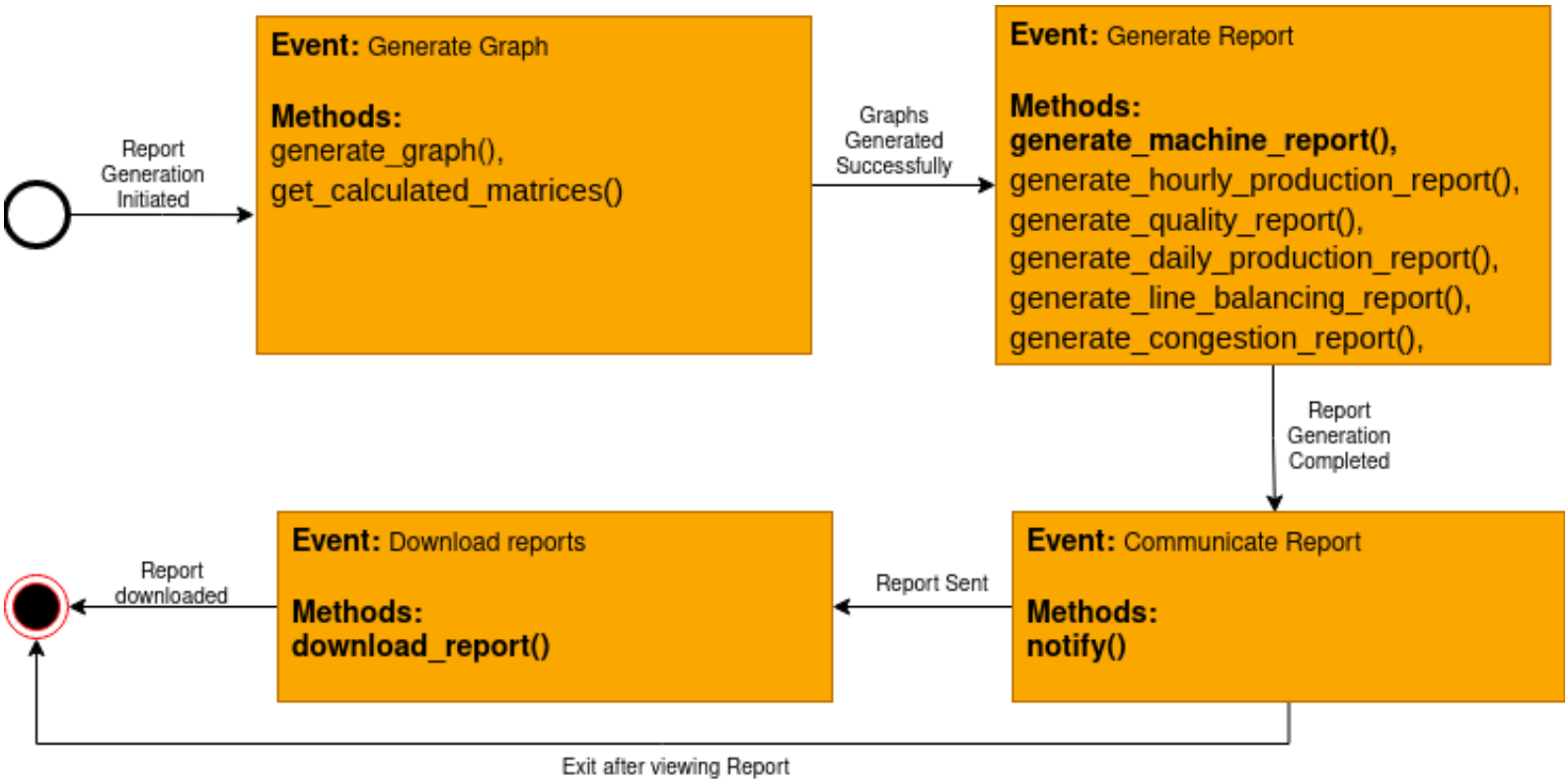


Figure 56: State transition diagram: Report

STD ID: 10

Class: Quality

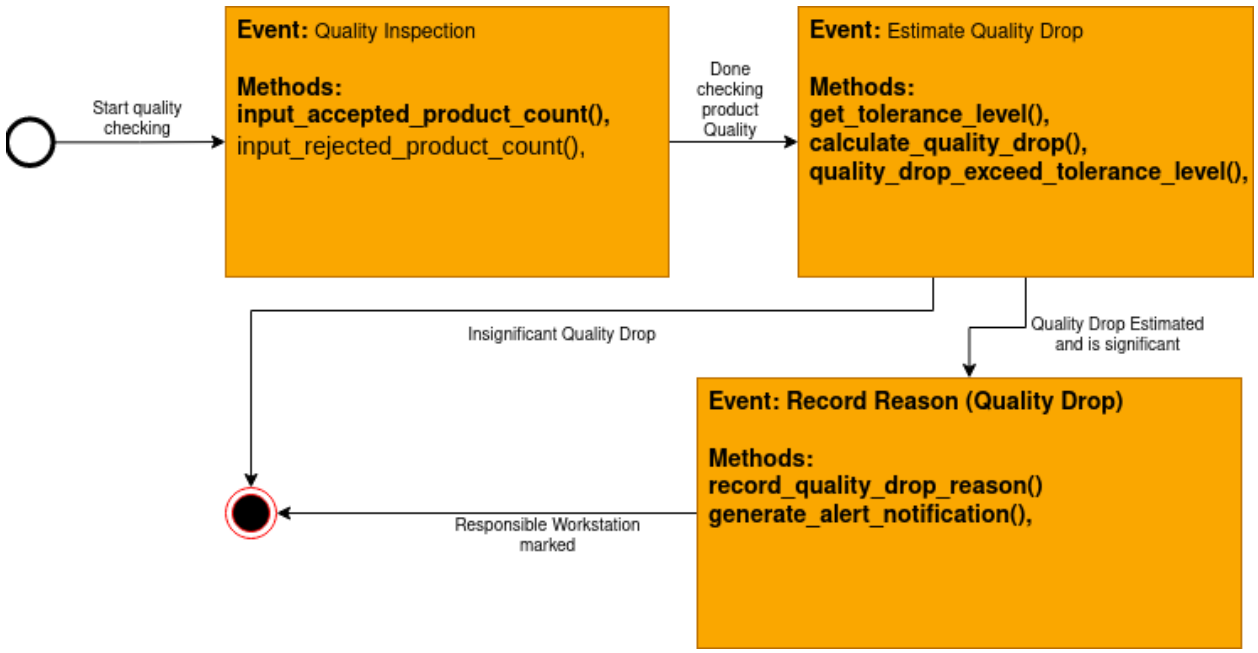


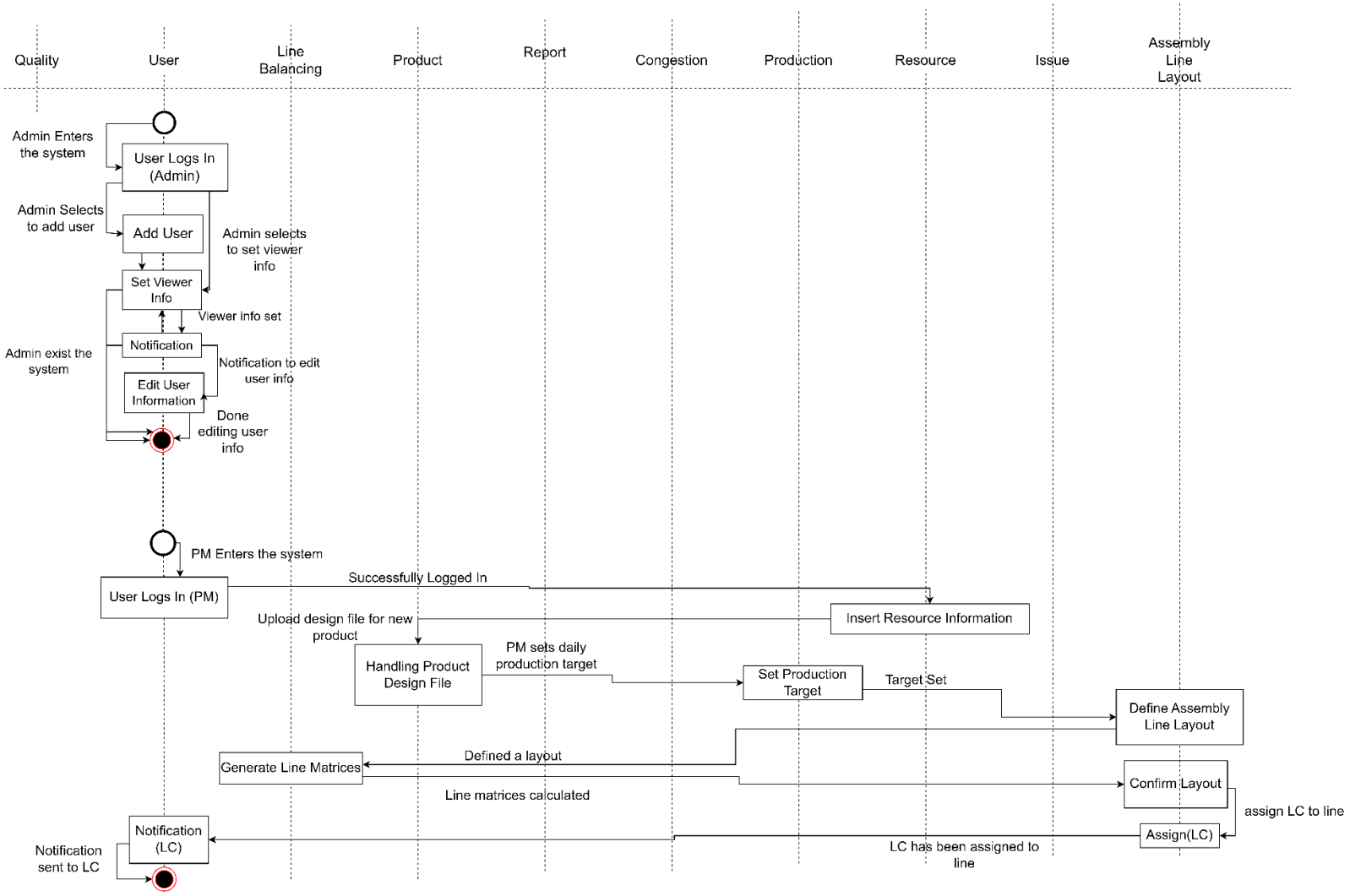
Figure 57: State transition diagram: Quality

7.3 Sequence Diagram

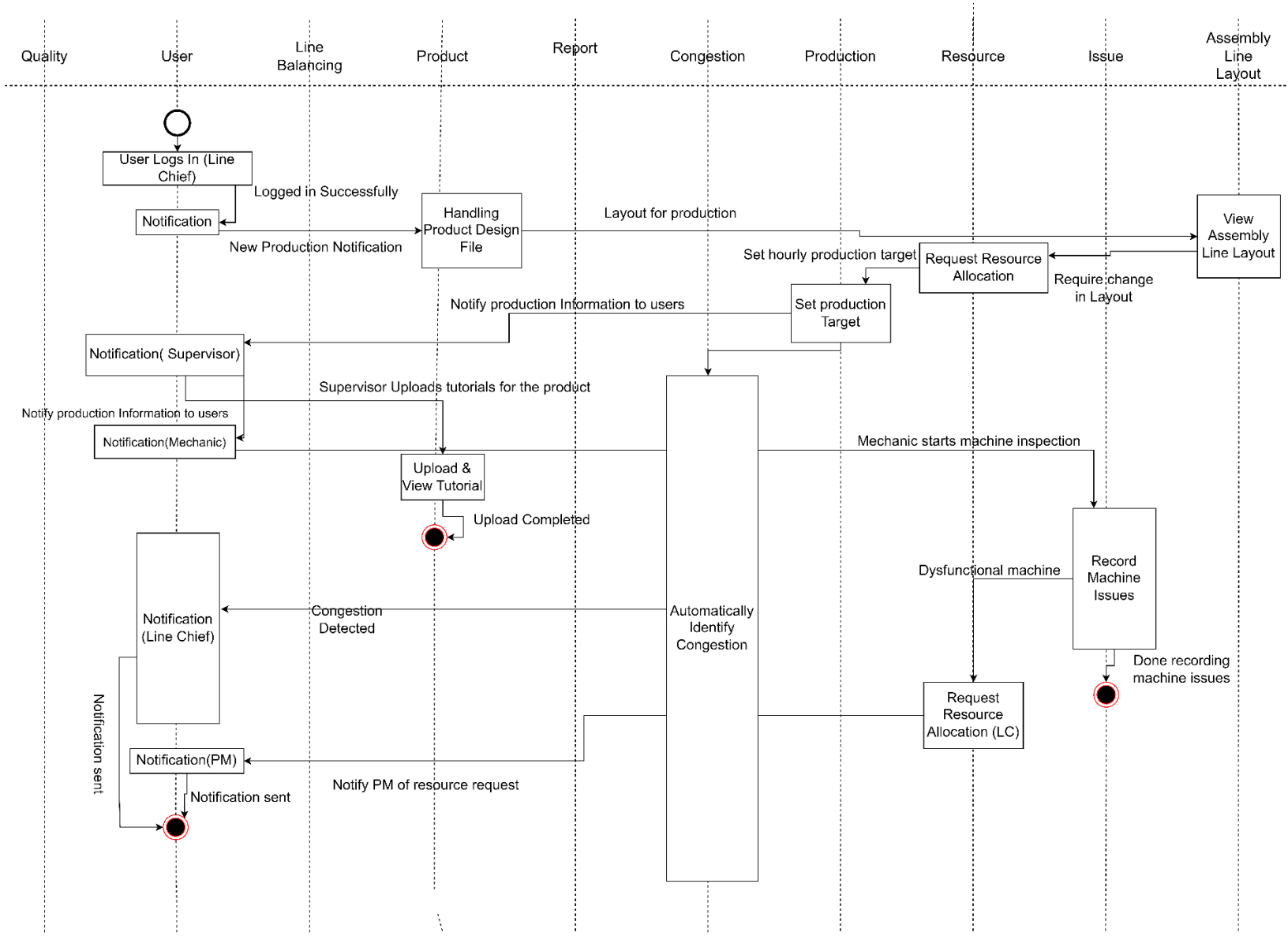
Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focused and they show the order of the interaction visually by using the vertical axis of the diagram to represent time, what messages are sent and when.

For better quality diagram enter this link:
https://app.diagrams.net/#G1JTuyo3z4pbgWdTPiwHHpO2JEkh2IlaGC

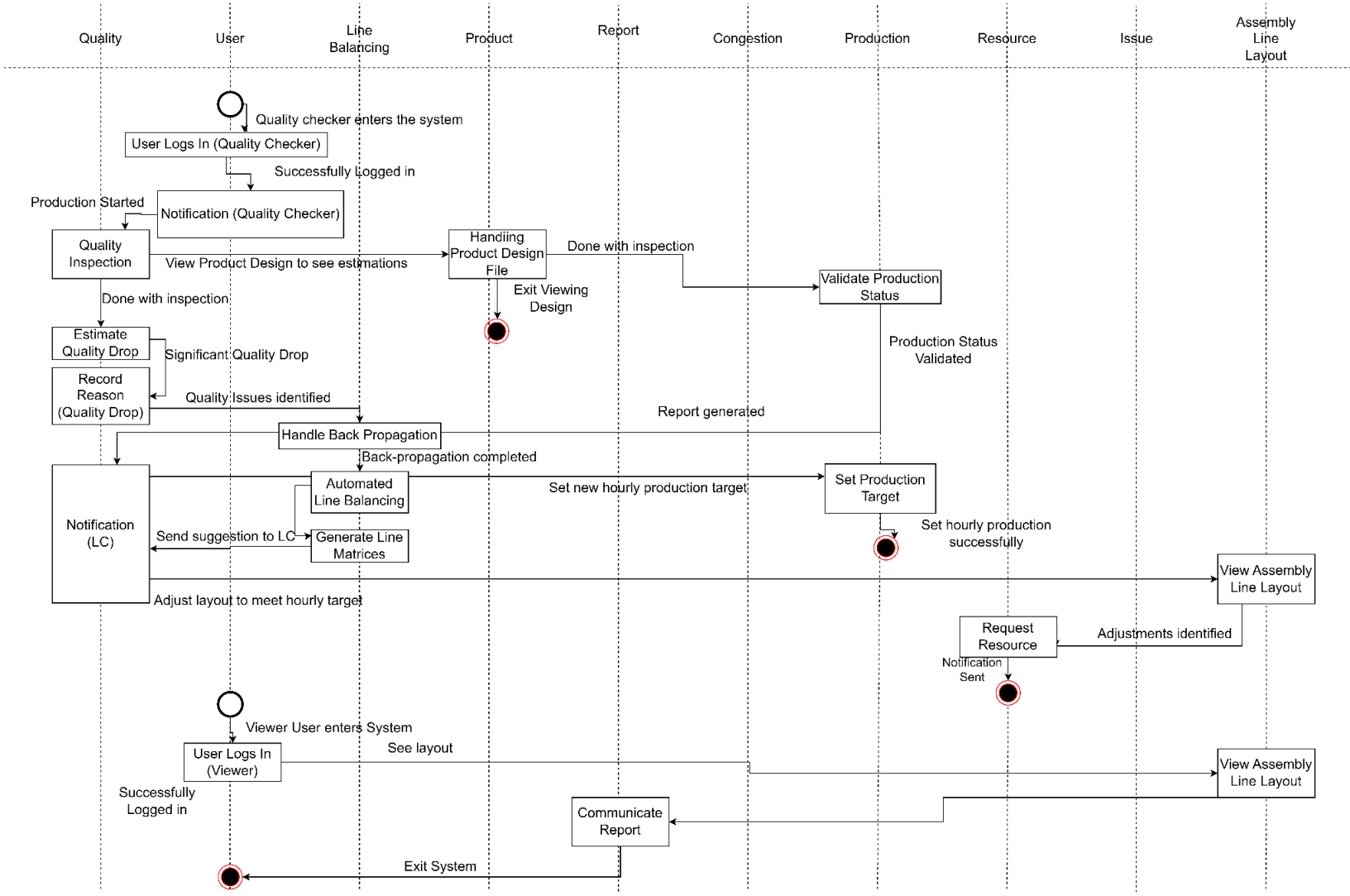
Login & Line Creation



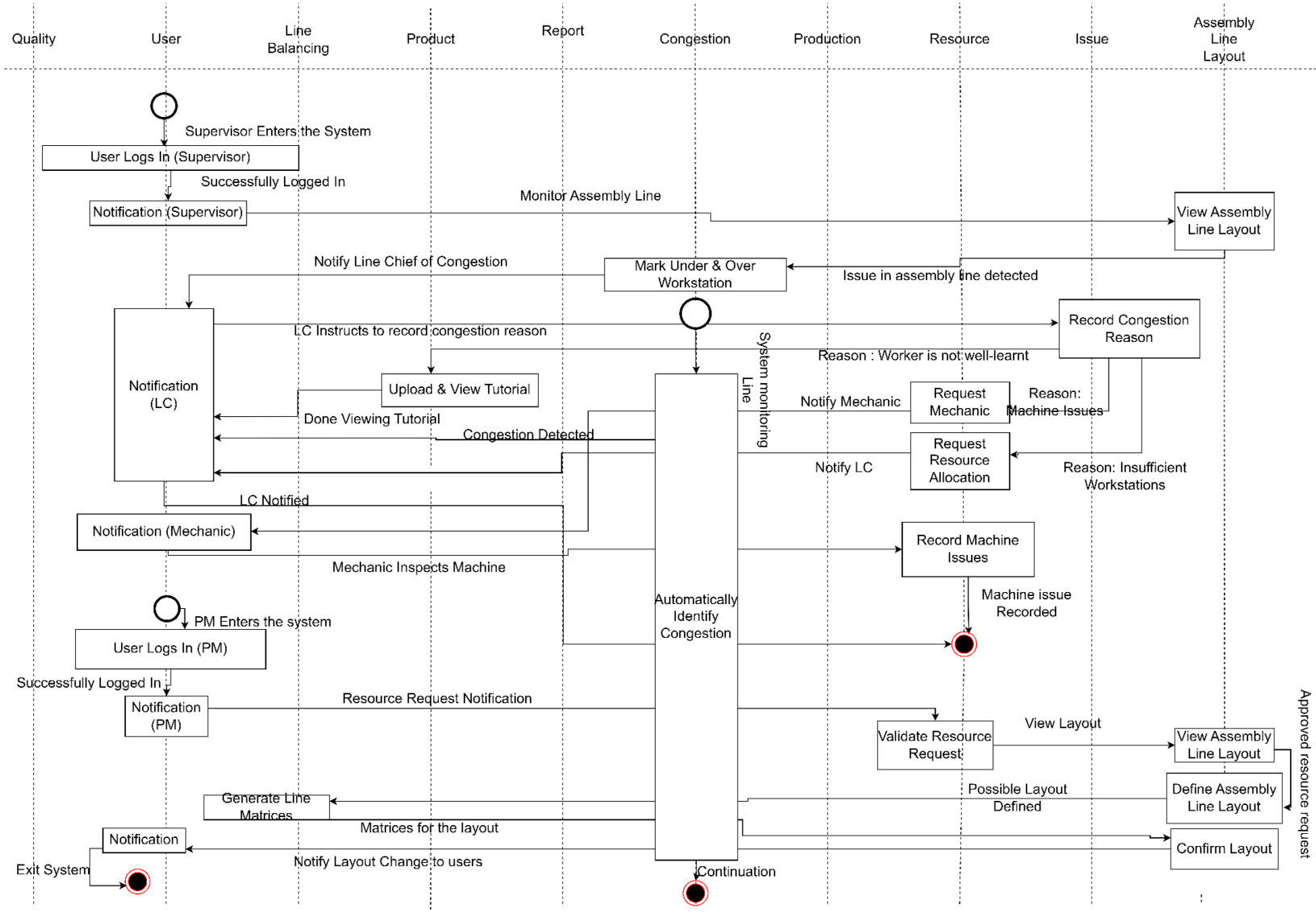
Production Starts



Production Occurring



Change Layout For Line Balancing



Report Generation

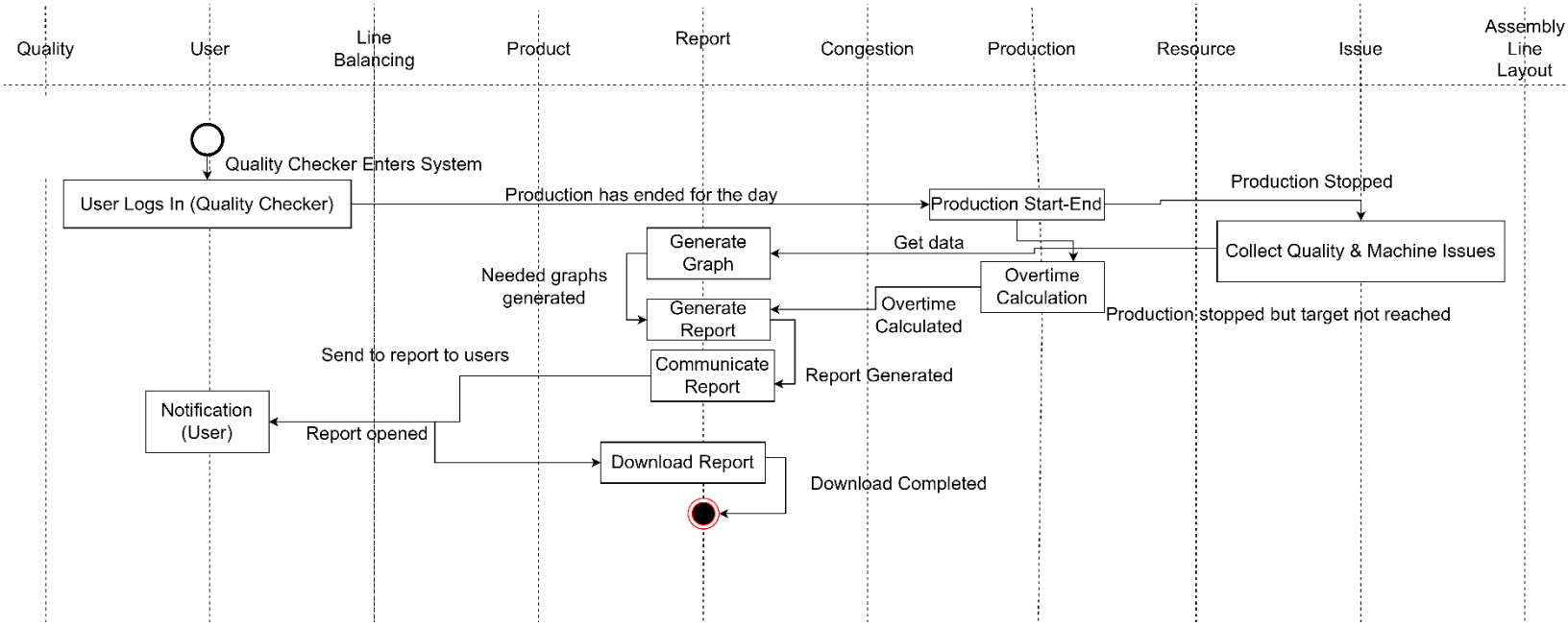


figure 58: Sequence diagram

References

[1]: Software Engineering A Practitioner's Approach 7th Edition - Roger Pressman