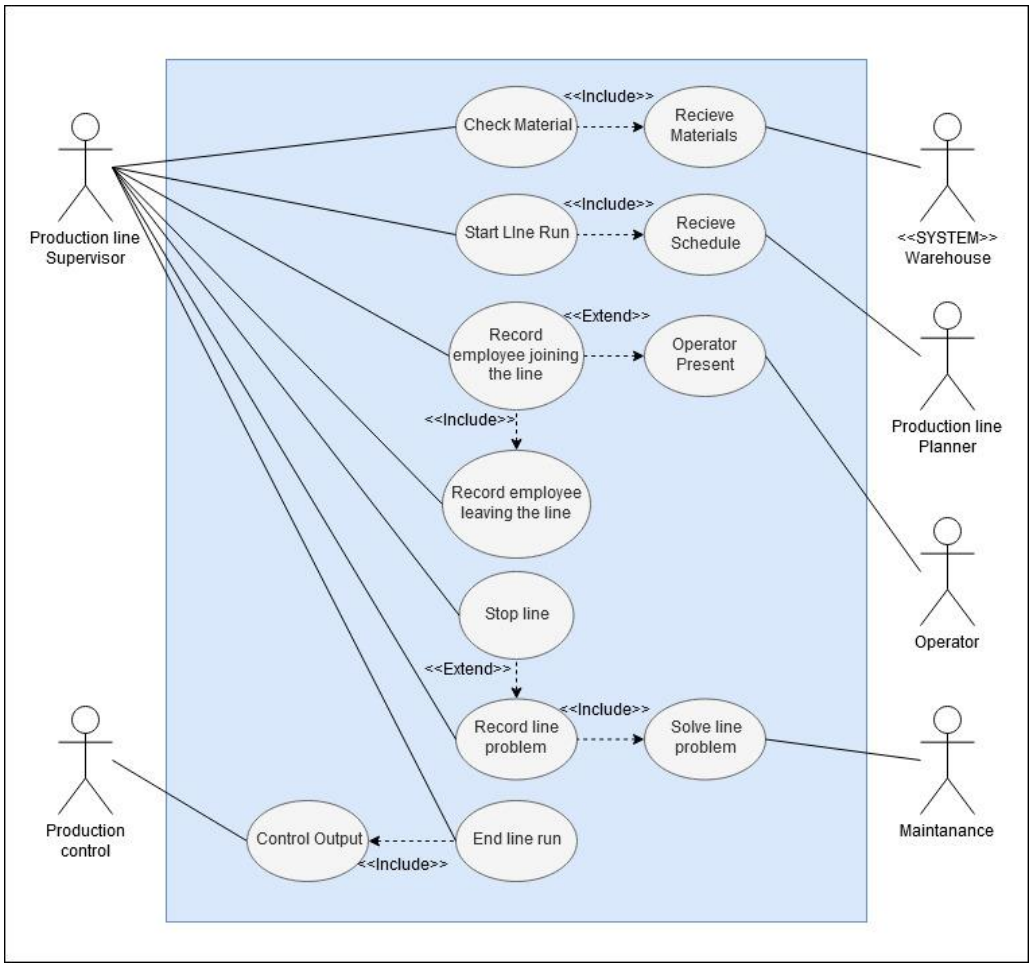
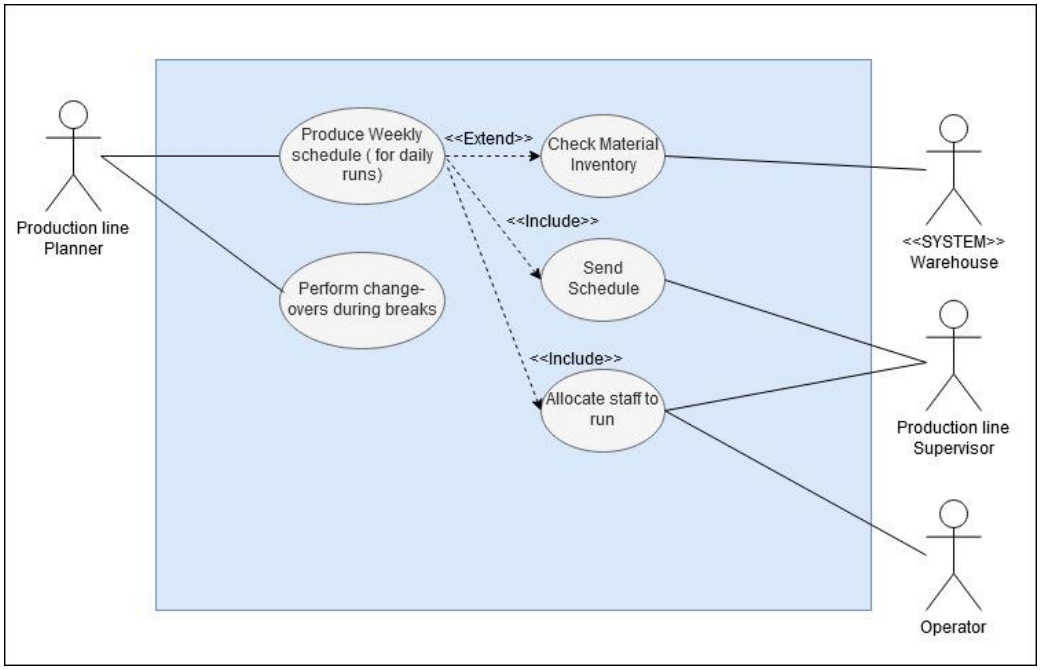
(A) Draw a Use-Case DIAGRAM which shows the functionality that the system should provide that is used by the PRODUCTION LINE SUPERVISOR.



(B) Draw a Use-Case DIAGRAM which shows the functionality that the system should provide that is used by the PRODUCTION PLANNER



(C) Write Use-Case DESCRIPTIONS for the following use-cases:

Title	START LINE RUN
Pre-conditions	Have required materials Staff present. Job card made from weekly shedule
Post-conditions	Line run starts without major issues.
Actors	Production line Supervisor, Operators.
Purpose	Staple to start daily run operations.
Description	If Staff and material is present in accordance to the job card the line can start
Alternative Flow	Lack of personnel: Slower production line. Lack of personnel: Acquire personnel from other lines.

Title	RECORD EMPLOYEE JOINING THE LINE
Pre-conditions	Employee present
Post-conditions	Employee's start-time added to production record sheet
Actors	Production line Supervisor, Operator.
Purpose	Record employee's start-time.
Description	Supervisor copy the job number from the job card and the current time onto a new timesheet for every new operator joining in the morning. If they join mid-way they might get one, otherwise this is done at the end of the day (Quite diffuse, therefore our diagrams make one every time a new operator joins the line).
Alternative Flow	-

Title	RECORD EMPLOYEE LEAVING THE LINE
Pre-conditions	Employee working at line leaves
Post-conditions	Employee's end-time added to timesheet
Actors	Production line Supervisor, Operator.
Purpose	Record employee's end-time.
Description	Supervisors writes down the end-time on the leaving operators Timesheet and deducts time if breaks totals more then 15 minutes.
Alternative Flow	-

Title	STOP LINE
Pre-conditions	Coffee break Or Meal-time break Maintenance issues Or Lack of materials End of run
Post-conditions	Line has been stopped
Actors	Production line Supervisor.
Purpose	Stop the line
Description	Either stop the line because of a planned break, issue or lack of materials.
Alternative Flow	1: Lack of materials: proceed to RECORD LINE PROBLEM 2: Issue: proceed to RECORD LINE PROBLEM 3: Planned break: STOP LINE 4: End of run: END LINE RUN

Title	RECORD LINE PROBLEM
Pre-conditions	STOP LINE: Alternative flow 1 or 2.
Post-conditions	Problem recorded.
Actors	Production line Supervisor.
Purpose	Record problem during certain time.
Description	Supervisor Contacts maintenance if the line breaks, warehouse or external suppliers if the materials runs out. Then they record the downtime while the line's not running and try to find useful things for the staff to do.
Alternative Flow	1: Lack of materials: Contact the warehouse or external sources. 2: Issue: Supervisor tries to solve it, contacts maintenance if required.

Title	END LINE RUN
Pre-conditions	Run completed.
Post-conditions	Daily run finished.
Actors	Production line Supervisor & Production Control
Purpose	End daily run.
Description	Supervisor notes the finish time on the production record sheet. Production Control verifies the quantity produced and note this on the production record sheet. Then the supervisor totals all the absences and totals the hours for each operator. If someone joined the line in mid-session, they might not have a timesheet, so one is made out now. Finally, they return unused ingredients to the warehouse, tidying up the line ready for the next run.
Alternative Flow	-

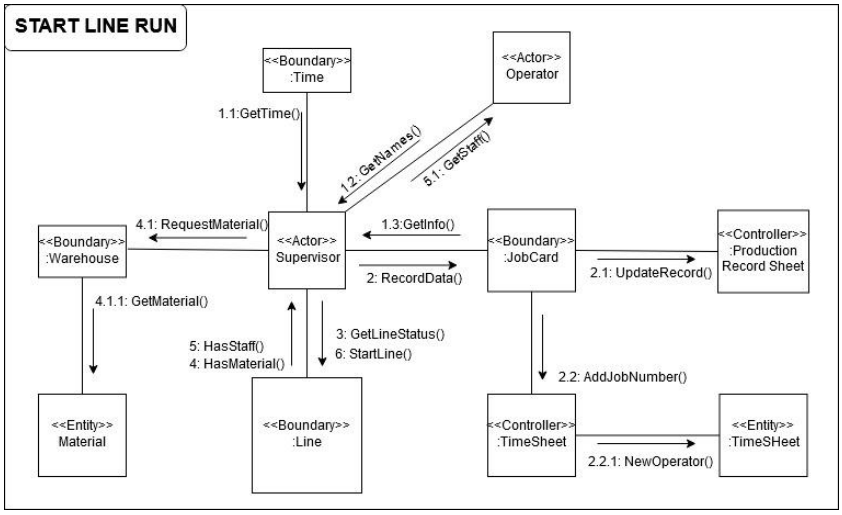
(D) Identify CANDIDATE CLASSES that can be derived from the transcript and the use cases. For each candidate class determine whether it is a BOUNDARY, CONTROL or ENTITY class.

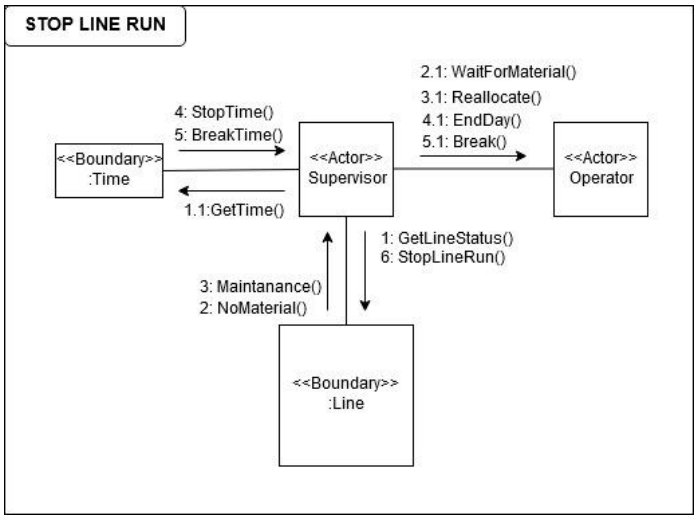
:Time <<Boundary>>
:TimeSheet <<Entity>>
:Production Record Sheet <<Boundary>>
:WareHouse <<Boundary>>
:Material<<Entity>>

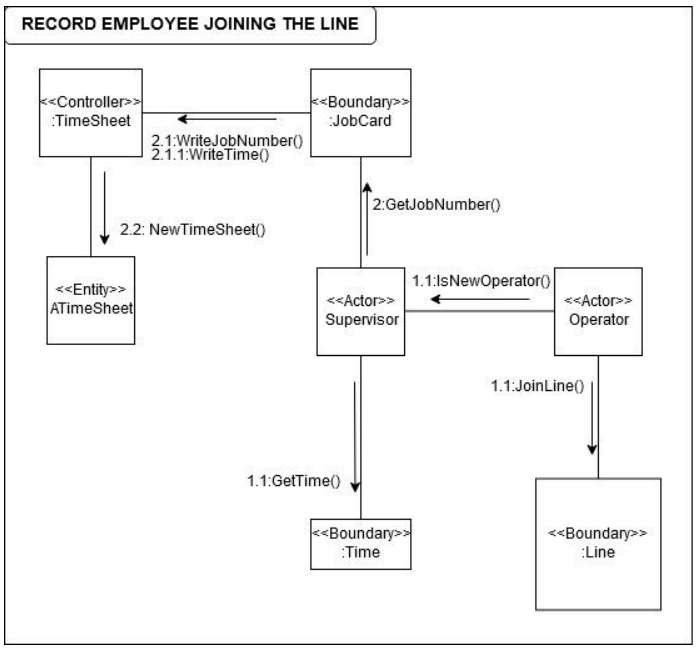
:JobCard<<Boundary>>
:Line <<Boundary>>
:Maintanance<<Boundary>>
:ExternalSupplier<<Boundary>>
:ProductionRecordSheet<<Controller>>

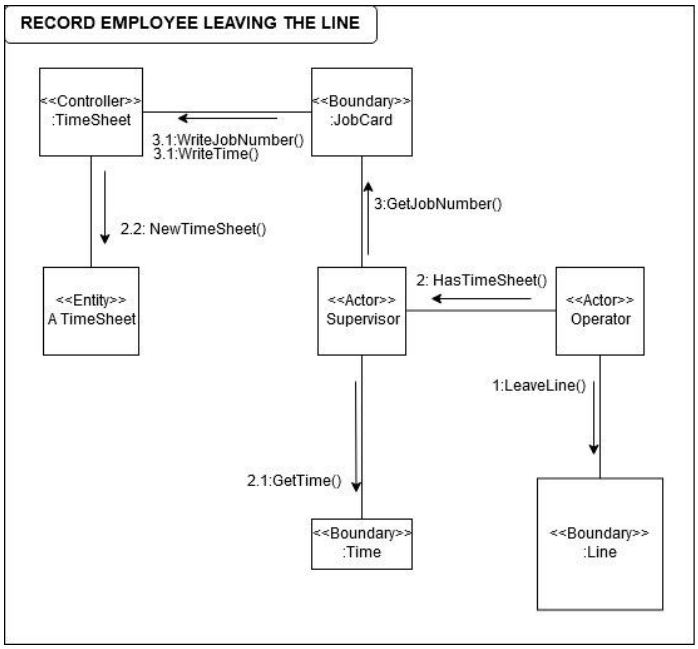
:Products<<Entity>> (new)

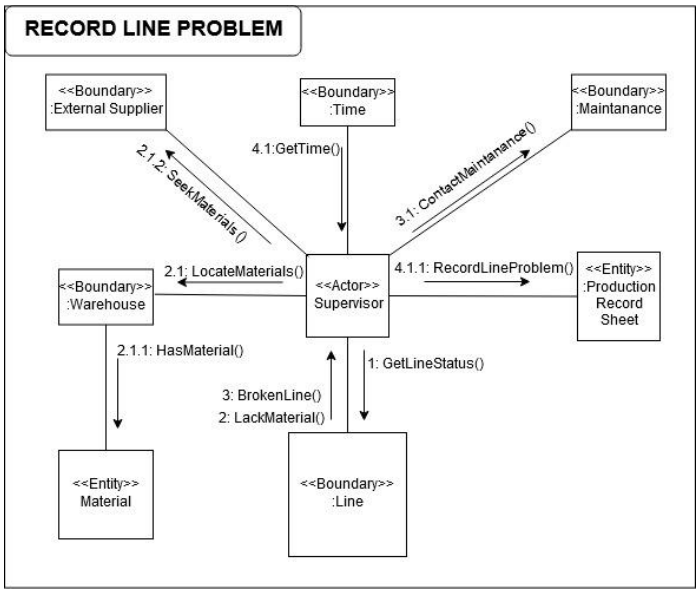
2. Communication diagrams [30%]
From these Use-Case DESCRIPTIONS and CANDIDATE CLASSES produce COMMUNICATION DIAGRAMS for EACH of the following use-cases:

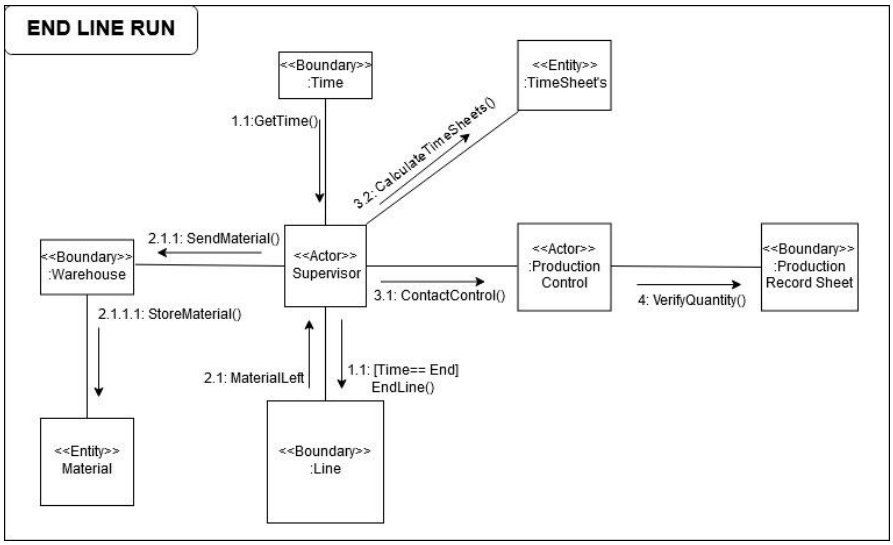






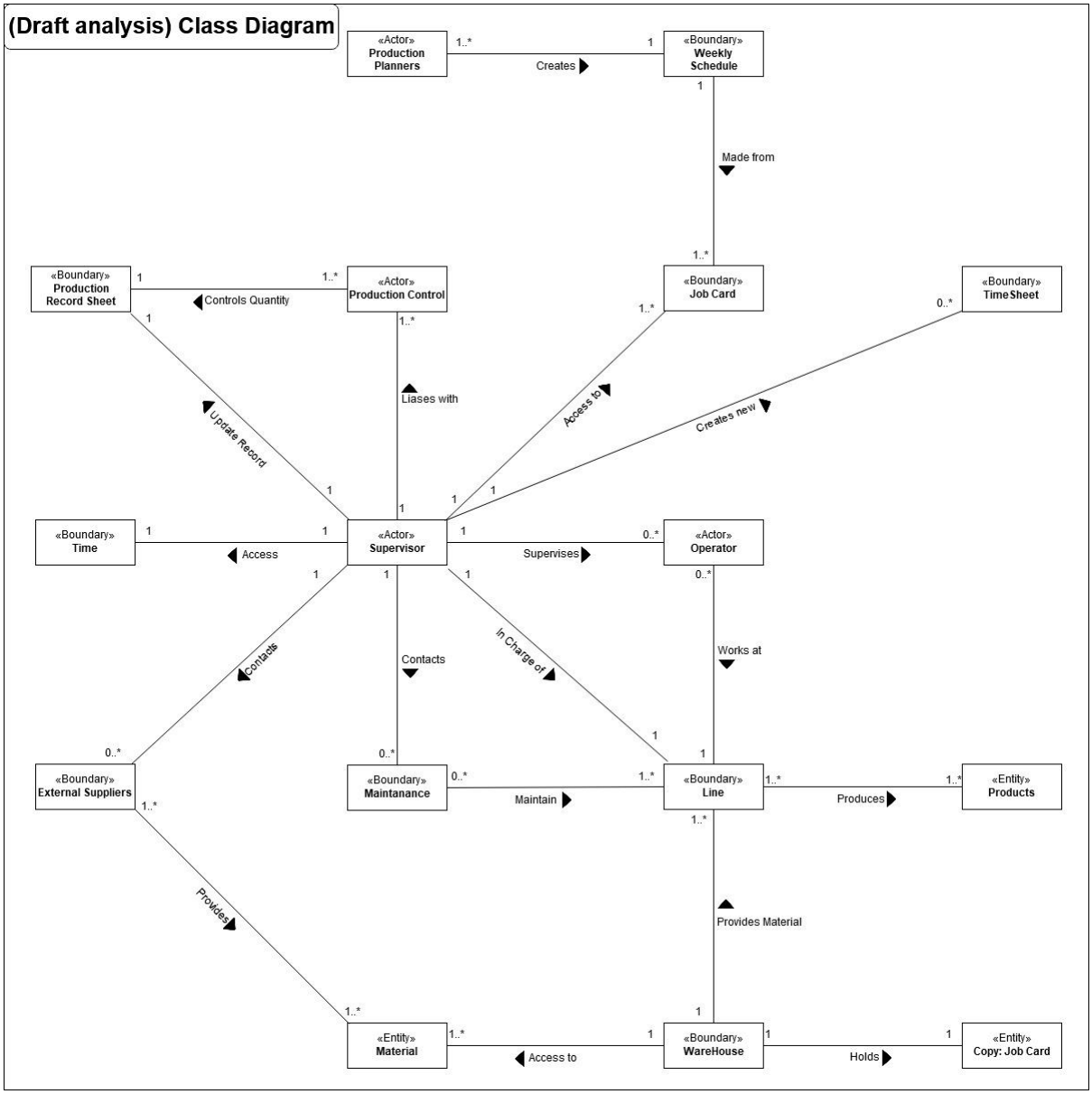

RECORD EMPLOYEE LEAVING THE LINE
<<Controller>>
:TimeSheet
<<Boundary>>
:JobCard
3.1:WriteJobNumber()
3.1:WriteTime()
2.2: NewTimeSheet()
3:GetJobNumber()
<<Entity>>
A TimeSheet
<<Actor>>
Supervisor
2: HasTimeSheet()
<<Actor>>
Operator
1:LeaveLine()
2.1:GetTime()
<<Boundary>>
:Time
<<Boundary>>
:Line


RECORD LINE PROBLEM
<<Boundary>>
:External Supplier
<<Boundary>>
:Time
<<Boundary>>
:Maintanance
2.1.2: SeekMaterials()
4.1:GetTime()
3.1: ContactMaintanance()
<<Boundary>>
:Warehouse
2.1: LocateMaterials()
<<Actor>>
Supervisor
4.1.1: RecordLineProblem()
<<Entity>>
:Production
Record
Sheet
2.1.1: HasMaterial()
1: GetLineStatus()
3: BrokenLine()
2: LackMaterial()
<<Entity>>
Material
<<Boundary>>
:Line

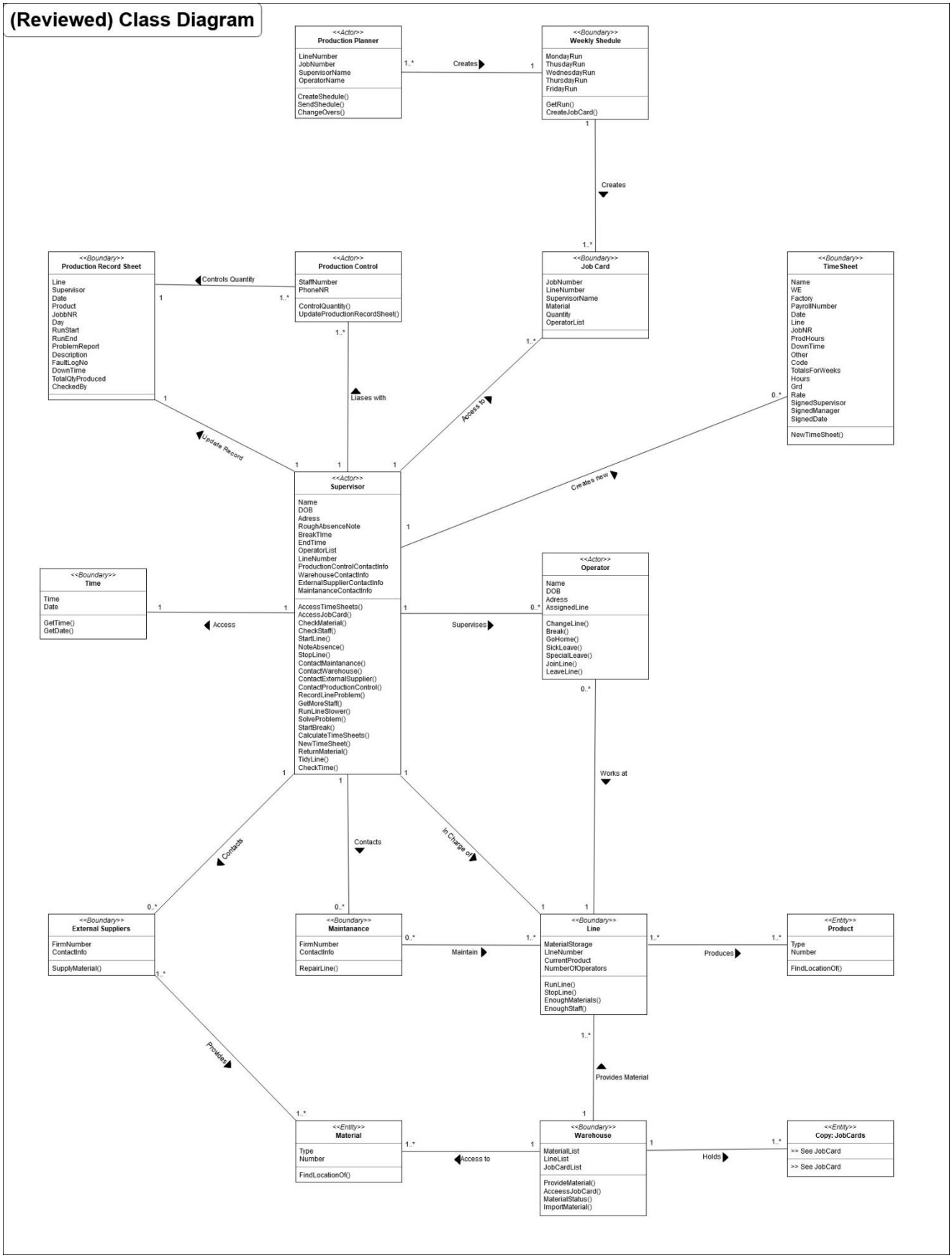

END LINE RUN
<<Boundary>>
:Time
<<Entity>>
:TimeSheet's
1.1:GetTime()
3.2: Calculate TimeSheets()
2.1.1: SendMaterial()
<<Boundary>>
:Warehouse
<<Actor>>
Supervisor
<<Actor>>
:Production
Control
<<Boundary>>
:Production
Record Sheet
3.1: ContactControl()
4: VerifyQuantity()
2.1.1.1: StoreMaterial()
2.1: MaterialLeft
1.1: [Time== End]
EndLine()
<<Entity>>
Material
<<Boundary>>
:Line

3. Class diagram [40%]

(A) First produce a draft analysis class diagram, initially showing only classes and associations.



(B) Then review your analysis class diagram and add any attributes and operations that you think are justified by your use cases. Make reasonable assumptions and add others that you think might be justified by other use cases not directly derived from the transcript.



(C) Write a short summary reviewing the communication and class diagrams. Particularly identify whether your diagrams are consistent with each other and the requirements analysis. Also describe whether you have had to impose additional requirements that were not originally provided in the transcript.
If this is the case, explain what exactly are these requirements, and why you have had to impose these additional requirements.

Comparing my communication- and class-diagrams I can see that they are consistent with the requirement analysis, but they aren't as complete.

Summarizing the six communication-diagrams we can see that the classes used are not entirely coherent with the full class-diagram. There are classes in the class-diagram that has not been brought up in the initial communication-diagrams. These include "Production planner", "Schedule", "Material" and "Products". These additional requirements have been added to showcase parts of the systems that wasn't encompassed within the use-cases given in exercise two. In other words, these six use-cases doesn't cover the entire system mentioned in the requirements.

"Material" and "Products" were added to specify what is being produced or passed between External suppliers, Warehouse and Line.

"Production planner" and "Schedule" was added as a top-level documentation that generates the Job cards used by the Supervisor and warehouse.

Some requirements have been given low priority in the class-diagram. Instead of being turned into a class they've been set as methods in another relevant class. Some examples of this is "RunLineSlower", "RoughTimeNotes", "SickLeave" and "SpecialLeave" for the Supervisor class.

In the communication-diagrams "People go missing", "Rough time notes", "Sick leave" and "Special leave" have been ignored because the very logic of joining or leaving a line includes these alternative ways to participate. The only effective difference, if these were to be taken into consideration, would be that a medical documentation would be passed to an unspecified entity and the total time calculation might be reduced.

More on UML-notation: I've seen class-diagrams with or without stereotypic identities. I've also seen UML-notations using <<Actor>> as a <<Entity>> which allows connections to stereotypes other than <<Boundary>>.

In order to keep these diagrams coherent, I've proceeded to use <<Actor, Boundary, Controller, Entity>> as a standard.

For graphical purposes a "copy: JobCards"<<Entity>> has been made as a reference to "Job Card"<<Boundary>> for Warehouse.