**“Software Engineering”**

**Course**

**a.a. 2017-2018**

**Template version 1.0**

**Deliverable #2**

**Lecturer: Prof. Henry Muccini (henry.muccini@univaq.it)**

**Industrial Robot dashboard**

|  |  |
| --- | --- |
| **Date** | <gg/01/2018> |
| **Deliverable** | <*Deliverable 3 - Finale>* |
| **Team (Name)** | Unnamed Group |

|  |  |  |
| --- | --- | --- |
| **Team Members** | | |
| **Name & Surname** | **Matriculation Number** | **E-mail address** |
| Santi Filippo | *232028* | *filippo.santi@student.univaq.it* |
| De Ciantis Stefano | *242860* | *stefano.deciantis@student.univaq.it* |
| Ubaldi Davide | *250668* | *davide.ubaldi@outlook.it* |

**Project Guidelines**[do not remove this page]*This page provides the Guidelines to be followed when preparing the report for the Software Engineering course. You have to submit the following information:*

* *This Report*
* *Diagrams (Analysis Model, Component Diagrams, Sequence Diagrams, Entity Relationships Diagrams)*
* *Effort Recording (Excel file)*

***Important:***

* ***document risky/difficult/complex/highly discussed*** *requirements*
* *document decisions taken by the team*
* ***iterate****: do not spend more than 1-2 full days for each iteration*
* ***prioritize*** *requirements, scenarios, users, etc. etc.*

Project Rules and Evaluation Criteria

***General information:***

* *This homework will cover the 80% of your final grade (20% will come from the oral examination).*
* *The complete and final version of this document shall be not longer than 40 pages (excluding this page and the Appendix).*
* *Groups composed of five students (preferably).*

*I expect the groups to submit their work through GitHub*

***Use the same file to document the various deliverable.   
Document in this file how Deliverable “i+1" improves over Deliverable “i".***

**Project evaluation:**

*Evaluation is not based on “quantity” but on “quality” where quality means:*

* *Completeness of delivered Diagrams*
* *(Semantic and syntactic) Correctness of the delivered Diagrams*
* *Quality of the design decisions taken*
* *Quality of the produced code*

Table of Contents of this deliverable

|  |  |
| --- | --- |
| Pagina | Contenuto |
|  |  |
| 5-6 | List of Challenging/Risky Requirements or Tasks |
| 7-8-9-10-11-12-13-14 | Requirements Collection ; FR ; Use Case ; Specific Requirements ; NFR ; Prioritization |
| 15-16 | Analysis Model ; Robustness Diagram ; Description of Robustness Diagram |
| 17-18-19-20-21-22 | Software Architecture ; Description of Component Diagram ; Sequence Diagrams |
| 23  24-25  26-27  28-29  30  31-32-33-34 | ER Design  Class Diagram ; Description of Class Diagram  Design Decisions  Explain how the FRs and NFRs are satisfied by design  Effort Recording  Code Prototype |

List of Challenging/Risky Requirements or Tasks

*<In this section, you should describe using the table below the most challenging or discussed or risky design tasks, requirements, or activities related to this project. Please describe when the risk arised, when and how it has been solved.>*

PLEASE FILL IN THIS TABLE AT EACH DELIVERABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **Challenging Task** | **Date the task is identified** | **Date the challenge is resolved** | **Explanation on how the challenge has been managed** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

A. Requirements Collection

*In this section, you should describe both the application* ***features/functional*** *requirements as well as the* ***non functional*** *ones. You shall also document* ***constraints*** *and* **rules***, if they apply.*

**Che cosa deve fare il sistema?**

Il sistema riceve in input una serie di segnali inviati da sensori presenti sui robot che si occupano del monitoraggio delle componenti hardware del robot stesso, i robot sono raggruppati in cluster che a loro volta sono raggruppati in aree, i robot che appartengono allo stesso cluster svolgono le stesse operazioni avendo il medesimo hardware. Ogni robot, cluster o area viene identificato univocamente attraverso un codice alfanumerico. Una volta memorizzati il sistema si occuperà di processare i dati calcolando l’Inefficiency Rate. Il sistema fornirà un’interfaccia (dashboard) che visualizza l’Inefficiency Rate di un cluster o di un singolo robot.

A.1 Functional Requirements

*<List, in bullet points, all the functional requirements your system shall implement>   
<Describe functional requirements and stakeholders through Use Case Diagrams>. <Then, prioritize them, and provide a table-based description of the most important requirements>*

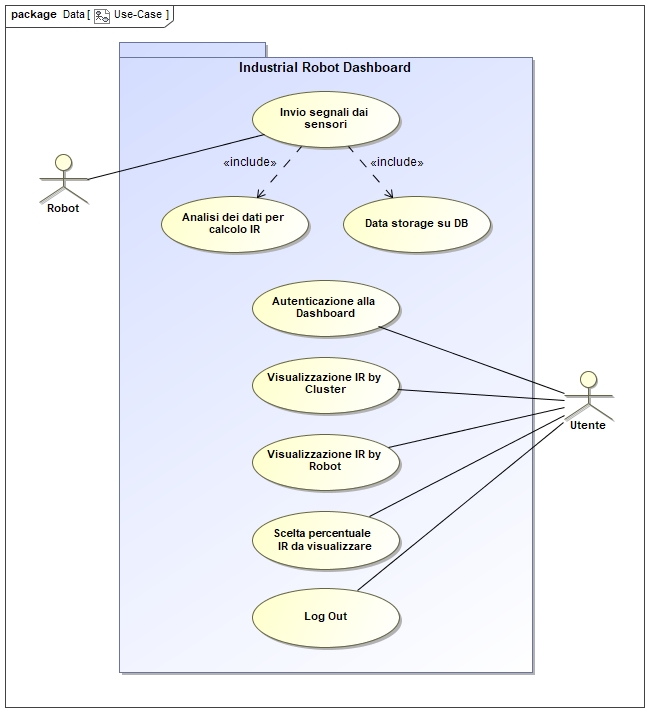
*<Provide a table-based description of the most important requirements, using the Alistair Cockburn Use Case Template>*

*PLEASE COPY HERE: (i) ALWAYS the diagram to be discussed, (ii) the text explaining the DECISIONS taken when creating the diagram (that is, do not spend time in EXPLAINING what the diagram says, but more on the decisions taken to create the diagram).*

PRIORITIZE THEM

FOR EACH DIAGRAM (Use Case D, Analysis Model, Class, Sequence) add a number and a label to it (e.g., Figure 1: Sequence diagram of the xxx scenario)

|  |  |
| --- | --- |
| Functional Requirement | Description |
| **Invio segnali** | Un robot dispone di una serie di sensori che inviano un flusso periodico di dati al sistema sotto forma di segnali, indicando se, in quel momento, per il robot è possibile eseguire una certa operazione o meno. |
| **Analisi real-time** | Il sistema effettua l’analisi real-time dei cambiamenti di stato trasmessi dai robot.  L’analisi real-time prenderà decisioni automatizzate sullo stato operativo dei robot, che può essere sostanzialmente riassunto in: produzione o manutenzione, a seconda dello stato di UP o DOWN dell’hardware relativo ai sensori. |
| **Calcolo IR** | Il sistema effettua l’analisi real-time dei cambiamenti di stato trasmessi dai robot.  L’analisi real-time prenderà decisioni automatizzate sullo stato operativo dei robot, che può essere sostanzialmente riassunto in: produzione o manutenzione, a seconda dello stato di UP o DOWN dell’hardware relativo ai sensori. |
| **Visualizzazione dati dashboard** | L'utente, attraverso la dashboard, deve essere in grado di visualizzare l’inefficiency rate di ogni robot da 2 punti di vista: del cluster in generale, oppure di un robot nello specifico.  La dashboard permetterà all’utente di selezionare il cluster o l’area di cui si vogliono visualizzare le informazioni. La dashboard mostrerà la percentuale di Inefficiency Rate per ogni cluster e per ogni robot con un colore diverso (rosso se Inefficiency Rate > al valore settato es:40%, verde altrimenti). |
| **Data Storage** | Il cambio di stato in input proveniente da ogni singolo robot viene memorizzato nel database. |



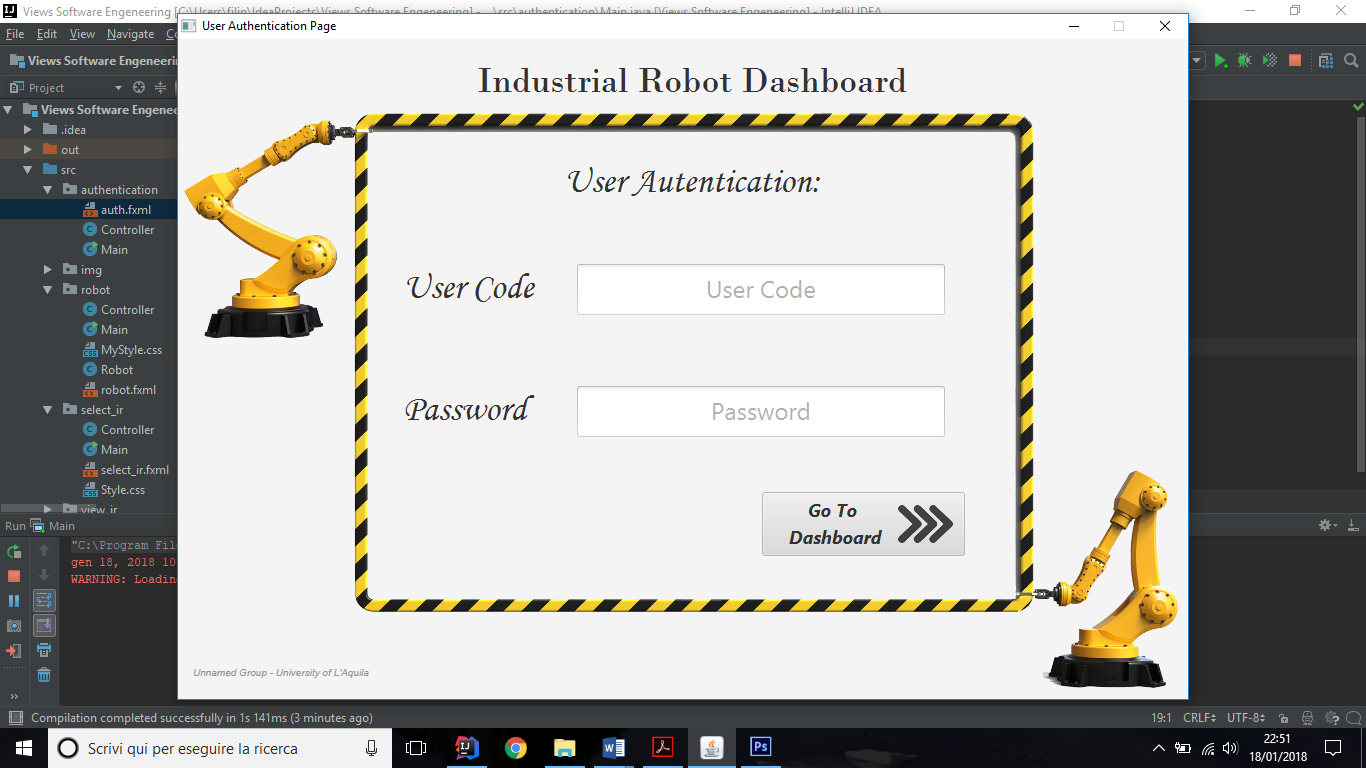
**Figura 1- Use Case**

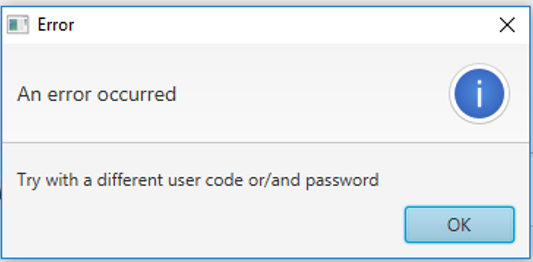
Abbiamo deciso di rappresentare *“Analisi dei dati per il calcolo IR”* e *“Data Storage su DB”* tramite la relazione *<<include>>* in quanto ci consente di completare lo use case precedente aggiungendo le logiche del calcolo dell’IR e dello storage dei dati al fine di fornire una panoramica completa su quello che è il ciclo dei segnali, dall’input alla memorizzazione.

Abbiamo, inoltre, deciso di rappresentare nel dettaglio tutte le azioni che un utente svolge o può svolgere in una sessione di lavoro standard, infatti l’ingegnere deve per prima cosa autenticarsi con successo per accedere ai dati di monitoraggio della dashboard, quando esso ha ottenuto l’accesso può decidere di accedere alle varie visualizzazioni dell’IR e ha la possibilità di usufruire di tutte le funzionalità della dashboard. Una volta svolto il lavoro può effettuare il log out e permettere ad altri utenti di usuruire di quel terminale.

A1.1 GUI Requirements (da riempire a partire dalla Versione 2)

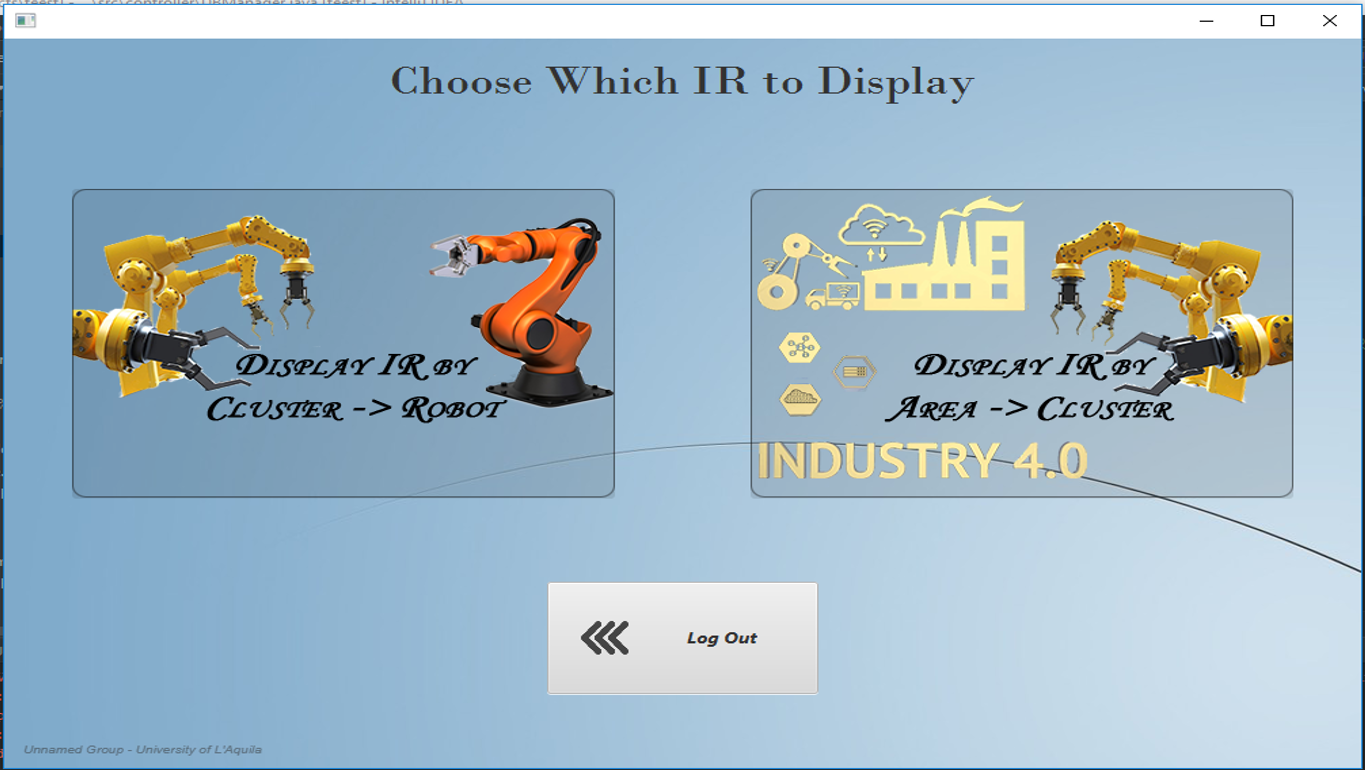
La GUI rappresenta il mezzo attraverso il quale il fruitore finale, ossia l’utente, si interfaccia con il sistema, nel nostro caso la GUI è rappresentata dalla DASHBOARD attraverso la quale l’ingegnere può monitorare lo stato di produttività di ogni singolo Robot da diverse “prospettive”.





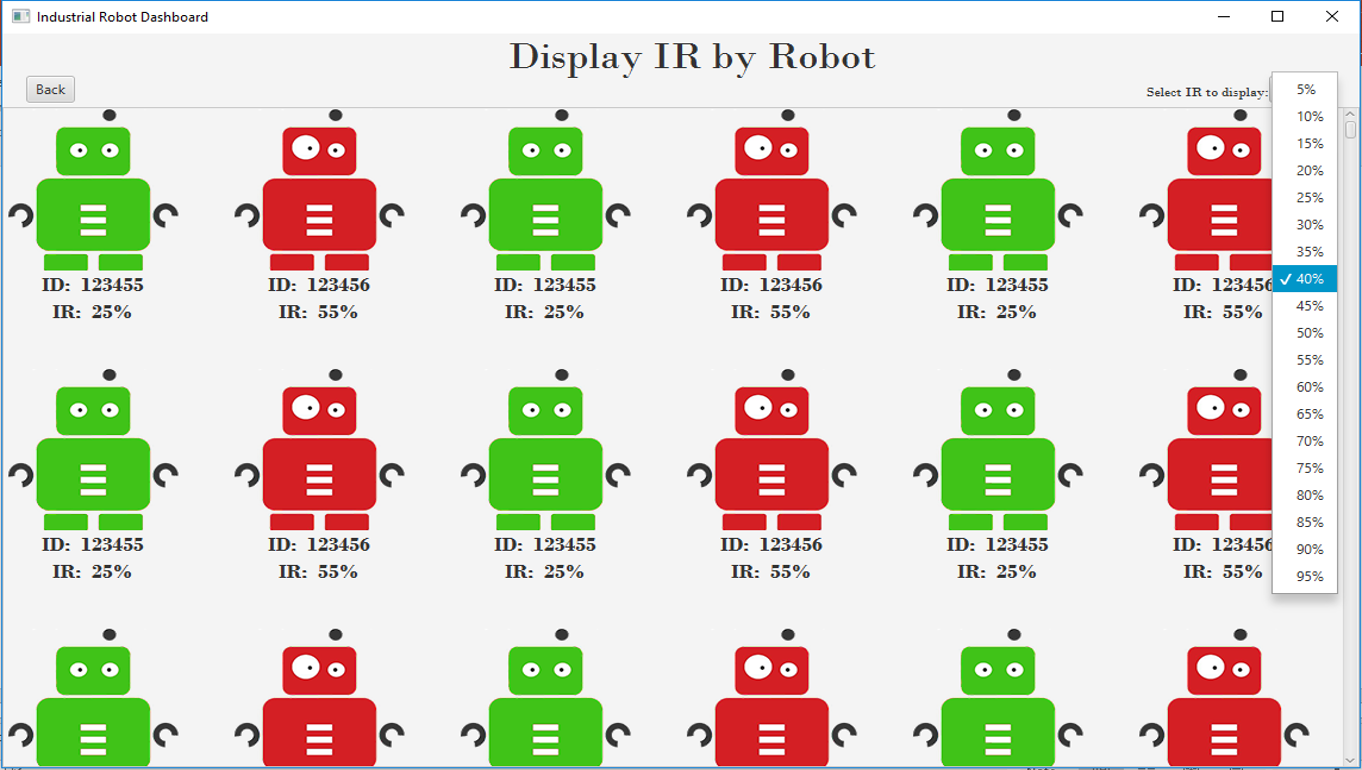
**A1.1.1 - Screenshot – Scenario Autenticazione + messaggio di errore**

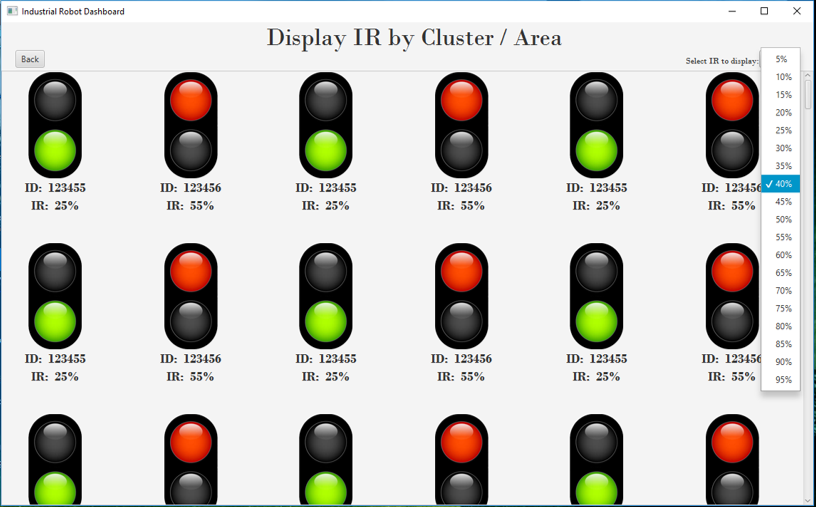
Nel momento in cui l’ingegnere decide di accedere alla Dashboard dovrà identificarsi nell’apposita finestra con i propri dati personali, solo nel caso in cui l’autenticazione avvenga con successo l’ingegnere può visualizzare i dati di monitoraggio (nel caso i dati non venano correttamente inseriti viene stampato un messaggio di errore e si dà possibilità all’utente di ritentare l’autenticazione).



**A1.1.2 - Screenshot – Scenario Scelta IR da Visualizzare**

Una volta autenticato l’utente ha modo di scegliere la “prospettiva” dalla quale vuole monitorare l’efficienza dei Robot tramite degli appositi bottoni. Questi permettono la visualizzazione dell’inefficiency rate (IR) di ogni singolo Robot, di ogni singolo Cluster oppure la massima astrazione si ottiene con il monitoraggio di un’Area. Per motivi di praticità e per facilitare la lettura dei dati all’utente finale si è scelto di raggruppare Cluster -> Robot e Area->Cluster in modo tale da avere una visualizzazione il più ordinata e precisa possibile (altrimenti visualizzare random circa 90'000 robot in un'unica finestra era forse più facile dal punto di vista implementativo ma poco utile allo scopo per cui la dashboard è stata realizzata) infatti scegliendo il bottone Cluster->Robot verranno visualizzati tutti i cluster con i relativi dati e cliccando su ognuno di essi sarà possibile visualizzare le informazioni di ogni robot presente all’interno, la stessa logica di navigazione è stata applicata all’altro bottone. L’utente avrà anche la possibilità di effettuare il log out per permettere a più ingegneri di utilizzare un singolo pc (es: cambio turno lavorativo).



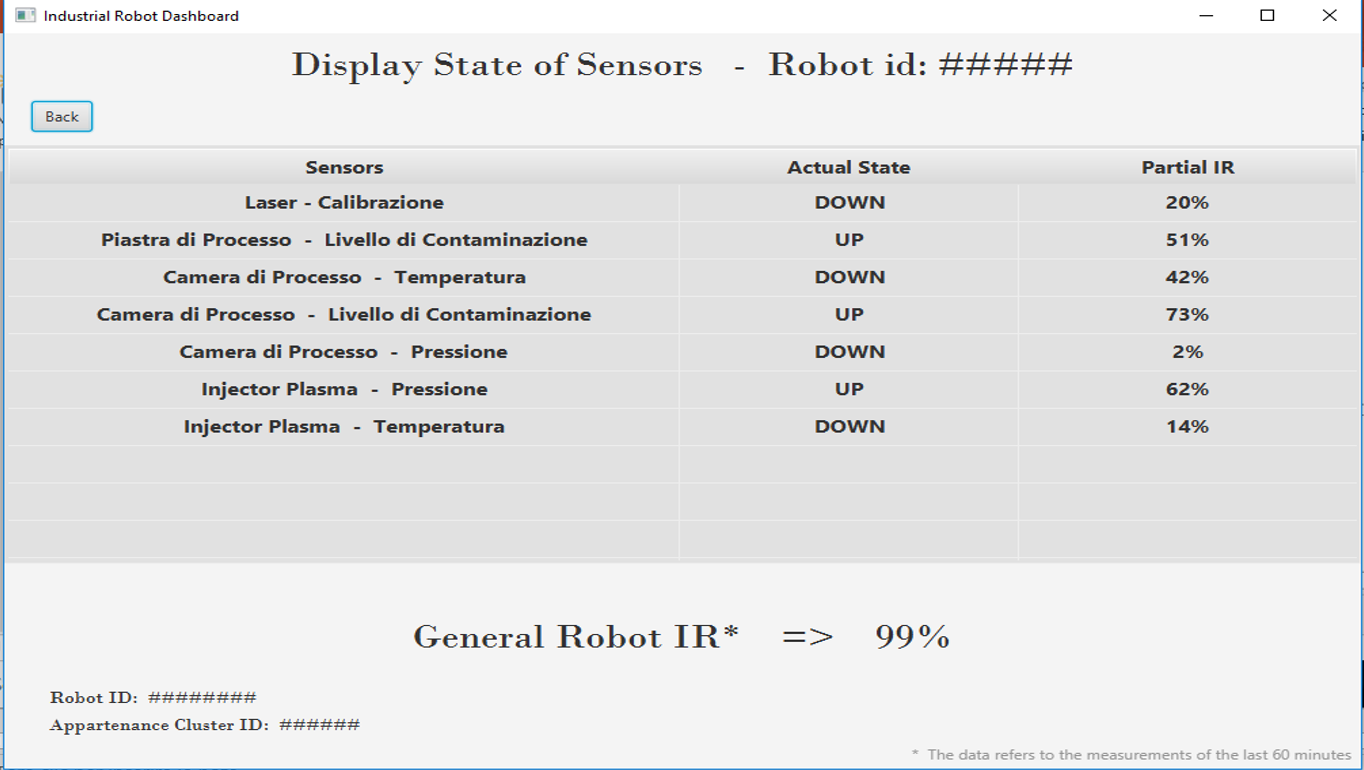
**A1.1.3.1 - Screenshot – Scenario Visualizzazione IR by Robot** 

**A1.1.3.2 - Screenshot – Scenario Visualizzazione IR by Cluser oppure by Area**

In questi scenari (**A1.1.3.1 e A1.1.3.2)** è possibile osservare nel dettaglio il compito che viene svolto dalla dashboard, ossia quello di monitorare e fornire dati all’ingegnere. Possiamo notare la presenza di 2 scenari differenti:

* + - **A1.1.3.1** è lo scenario rappresentante la visualizzazione by robot alla quale arrivo solo dopo aver navigato nei cluster, qui sono presenti i dati di monitoraggio di tutti i robot appartenenti al cluster precedentemente selezionato, sarà possibile fare click su ogni robot visualizzato per espanderne tutti i dettagli.
    - **A1.1.3.2** è lo scenario rappresentante la visualizzazione by Cluster o Area, anche qui, come si può intuire, la lettura dei dati è facilitata dai colori utilizzati, rosso o verde secondo il caso, nella visualizzazione by area sarà possibile cliccare su ogni area presente in modo da espandere e controllare i cluster che la compongono.

E’ inoltre importante permettere all’utente di personalizzare quanto più possibile la ricerca permettendogli di eseguire un settaggio secondo parametri personali, infatti, tramite l’apposita funzione presente in ciascuna delle finestre interessate, può decidere di cambiare il valore della percentuale di IR da evidenziare oppure di lasciare il valore di default, tutti i robot o cluster che avranno un valore di IR > o = al valore settato dall’ingegnere verranno contraddistinti in rosso, in modo da rendere immediata la comprensione del risultato, i valori < a quello settato saranno visualizzati in verde.



**A1.1.4 - Screenshot – Scenario Visualizzazione Dettagliata Singolo Robot**

Per permettere all’ingegnere di apprendere quante più informazioni e dati relativi ad ogni singolo robot è stata progettata una finestra che visualizza nel dettaglio tutte le informazioni che si possono ottenere da ogni robot, ossia: lo specifico dei 7 sensori che riportano statistiche aggiuntive non visualizzabili altrimenti (es: IR di ogni singolo segnale calcolato negli ultimi 60 minuti), lo stato attuale relativo ad ognuno dei 7 sensori, oltre ai dati relativi all’id del robot, al cluster di appartenenza e all’IR totale. Per accede a questa view è necessario navigare nei cluster scegliendo quello che ci interessa facendoci click spra, a quelpunto verranno visualizzati tutti i robot che lo compongono, facendo click su uno specifico robot si visualizzerà la finestra in questione.

A1.2 Business Logic Requirements (da riempire a partire dalla Versione 2)

La logica di business presente nel nostro sistema ruota tutta intorno alla manipolazione dei dati ottenuti in input, ovvero ai cambi di stato dei robot, che verranno processati dopo essere stati estrapolati dal database. La Business Logic infatti svolge i seguenti compiti:

* Eseguire i controlli sul DB per verificare eventuali cambi di stato tra i segnali ottenuti in ingresso e nel caso sovrascriverne il valore attuale;
* Eseguire i get() di tutti i valori dei segnali di tutti i robot per poter calcolare l’IR a 3 diversi livelli di astrazione (IR by Robot, IR by Cluster, IR by Area).
* Fornire i dati ottenuti in output sulla la GUI attraverso la quale l’utente si interfaccia al sistema per ottenere le informazioni che cerca;
* Rieseguire tutti i passaggi precedenti a distanze di tempo regolari in modo da garantire dati il più possibile freschi e aggiornati all’utente.

A1.3 DB Requirements (da riempire a partire dalla Versione 2)

Il DB memorizza i dati provenienti dai robot durante un intervallo di tempo prestabilito, tenendo uno “storico” dei segnali di cambio stato ricevuti, consentendo in seguito di effettuare le operazioni necessarie al calcolo dell’IR.

Lo storage dei dati avviene attraverso una componente server dedicata, che ‘intercetta’ i segnali inviati dai robot ed effettua le operazioni di inserimento nel database. Sul database sarà memorizzato solo l’ultimo cambio di stato il tutto sovrascritto al relativo dato precedentemente memorizzato.

Oltre che ai segnali inviati dai robot, nel DB teniamo traccia anche del timestamp: dato di fondamentale importanza, il cui scopo è facilitare i calcoli relativi al periodo di inefficienza di un robot, e quindi semplificare il processo tramite cui si otterranno i valori di IR relativi ai cluster e alle aree.

A.2 Non Functional Requirements

*<List and describe here the* ***most important*** *non functional requirements.>   
BE CAREFUL NOT TO MAKE CONFUSION AMONG DIFFERENT NON FUNCTIONAL REQUIREMENTS. PLEASE FOCUS ON NO MORE THAN THE 3-4 MOST IMPORTANT NON FUNCTIONAL REQUIREMENTS-*

|  |  |
| --- | --- |
| Non Functional Requirement | Description |
| Usability | Bisogna coinvolgere gli utenti nel processo di progettazione ed eseguire test frequenti al fine di raccogliere feedback. Buona parte del giudizio sulla usability sarà determinato dall’utente in base alla qualità dell’interfaccia finale (in questo caso la dashboard). |
| Reliability | Il sistema deve essere sempre pronto a ricevere dati dai sensori dei robot e quindi rimanere costantemente online (24/7, 365gg/anno), nel nostro caso specifico si necessita di un minimo di up time pari al 98%.  Inoltre deve fornire misurazioni costanti cercando, a contempo, di contenere il più possibile i costi, nel nostro caso è importante la precisione nel “timestamp” del cambio di stato ma non è richiesta particolare precisione nel valore del segnale essendo esso un semplice boolean. |
| Performance | Il sistema deve garantire dei tempi di risposta ottimali, più precisamente dovrà soddisfare 90.000 richieste al minuto provenienti da robot, senza aumentare eccessivamente i costi e/o i tempi di sviluppo. |
| Efficiency | Nel nostro caso l’architettura del sistema dovrà gestire fino a 90'000 robot, di conseguenza, supportare lo storage e il processing di almeno 90'000 messaggi al minuto.  La dashboard sarà aperta su almeno 100 PC in linea e dovrà aggiornarsi automaticamente al massimo ogni 5 minuti. |

A.3 Content

*<Describe the* ***data provenance*** *(use of external API, web service, DB …)>*

I dati vengono memorizzati in un database integrato nel Sistema, le tecnologie utilizzate sono MongoDB per quanto riguarda la base di dati e RoboMongo come DBMS.

A.4 Assumptions

*<Briefly document, in this section, the most relevant requirement assumptions/decisions you had to made during your project>*

* Viene dato per assunto che chiunque abbia progettato e/o scelto la parte software dei PC che verranno utilizzati dagli ingegneri e quindi i PC sopra i quali dovrà girare la nostra dashboard siano muniti di una versione del JDK pari a 1.8.0\_152 o più recente per la corretta visualizzazione delle finestre e la corretta funzione della logica Java.
* Va inoltre dato per assunto che il sistema sia debitamente collegato in rete con ogni robot o dispositivo di ricezione segnali (quello che per noi è il Server) preferibilmente con una connessione cablata di tipo Gigabit Ethernet LAN in modo da garantire efficienza e velocità elevata.
* Abbiamo inoltre assunto la presenza di un server che funge sia da ricevitore dei segnali sia da bus dati per il trasferimento degli stessi al database ce ne esegue il relativo storage. Il Server si occupa solo della ricezione e salvataggio e NON dell’analisi dei dati.
* Al momento del primo avvio i robot dovranno avere una produttività del 100% ossia tutte le partizioni hardware dei robot devono rientrare nelle specifiche e quindi essere in fase di UP in quanto al primo arrivo i segnali saranno impostati di default a 1.

***A.5 Prioritization***

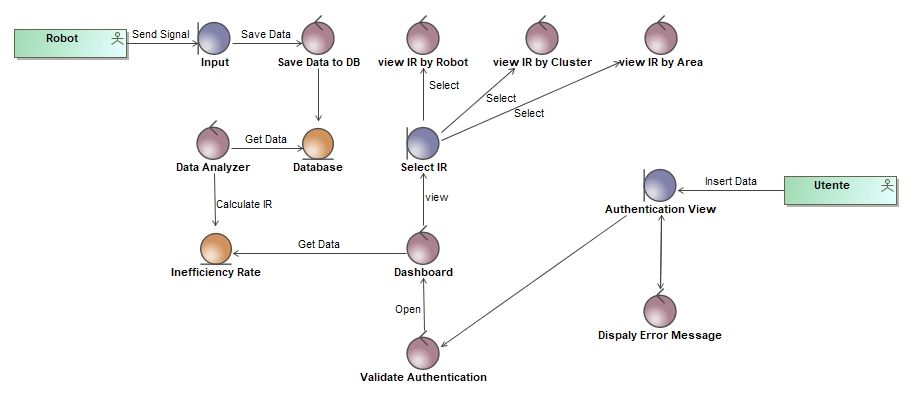
*Requisiti listati secondo ordine di priorità.*

|  |  |  |
| --- | --- | --- |
| Requisiti | Priorità | Commenti |
| **Invio dei dati** | Alta | Funzione indispensabile al corretto funzionamento dell'intero sistema |
| **Data storage** | Alta | È importante che i dati rimangano memorizzati sul database. Altrimenti sarebbe impossibile effettuare le operazioni di calcolo relative all'IR. |
| **Analisi real-time** | Media | Funzionalità indispensabile per l'analisi dei dati |
| **Calcolo IR** | Media | Funzionalità indispensabile per la produzione dell'inefficiency rate. |
| **Visualizzazione dati** | Media | Richiede che tutte le funzionalità precedenti siano eseguite correttamente, e consente all'utente di rapportarsi col sistema. |

B. Analysis Model

*<In this section, you shall report the Analysis Model produced for your system>*

*PLEASE REPORT: (i) ALWAYS the diagram to be discussed, (ii) the text explaining the decisions taken to create the diagram (that is, why did you create a certain analysis model design?)*



**Figura 2- Robustness Diagram**

Analizzando quindi gli Use Case generati in fase di collezione dei requisiti individuiamo le seguenti classi suddivise nelle tre tipologie previste da UML:

*Boundary*: Input, Select IR, Authentication View

*Controller*: Save Data to DB, Data Analyzer, Dashboard, View IR by Robot, View IR by Cluster, View IR by Area, Display Error Message, Validate Authentication

*Entity*: Data, Inefficiency Rate

L’utente si interfaccia con la schermata iniziale immettendo i dati di autenticazione, identificata da un oggetto boundary chiamato “Authentication View”. Da qui, esitono due possibilità, la prima, rappresentata dall’oggetto di tipo controller “Display Error Message” rappresenta il caso in cui l’autenticazione non vada a buon fine e quindi mostra a schermo un messaggio di errore eseguendo un refresh della pagina per dar modo all’utente di riprovare ad inserire i dati.

Nel secondo caso invece la fase di login è stata eseguita con successo attraverso l’oggetto di tipo controller “Validate Authentication” raggiungendo quindi la dashboard. Qui l’utente ha la possibilità di scegliere due modalità di visualizzazione relative all’inefficiency rate: IR by Robot, IR by Cluster oppure IR by Area.

Un robot ha la possibilità di interfacciarsi al sistema attraverso l’oggetto boundary “Input”. Il controller “Save to Database” si occupa dello storage dei dati ricevui all’interno della entity “Database”.

Da questo punto, il controllo passa al controller “Data Analyzer”, che recupera i dati necessari e li elabora, effettuando così il calcolo dell’IR. I risultati saranno poi presi dall’oggetto di tipo entity “Inefficiency Rate”, pronti per essere recuperati dal controller “Dashboard”.

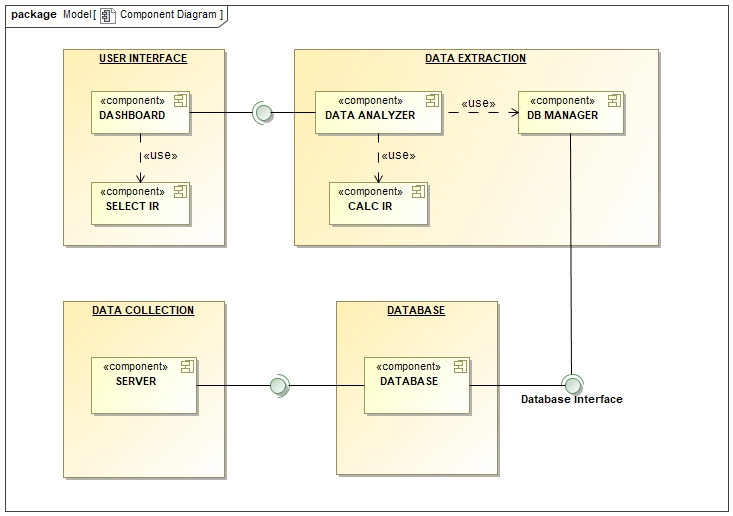
C. Software Architecture   
*<Report here both the static and the dynamic view of your system design, in terms of a Component Diagram, Class Diagrams and their related Sequence Diagrams >*

C.1The static view of the system: Component Diagram

*AVOID TO MAKE IT TOO COMPLEX AND FINE GRAINED. FOCUS MORE ON “HOW”*

*THE COMPONENTS SHALL INTERACT IN ORDER TO SATISFY THE REQUIREMENTS*

*ADD INTERFACES and their parameters*



**Figura 3- Component Diagram**

Una delle componenti del nostro sistema, ossia l’USER INTERFACE, si interfaccia con la componente relativa alla DATA EXTRACTION che attraverso il Data Analyzer recupera i dati dal database per processarli. Il DATABASE riceve i dati della componente Input relativa alla quarta componente DATA COLLECTION che ha come unico compito quello di memorizzare i dati ricevuti in input. Possiamo quindi distinguere il nostro sistema i 4 macro aree:

1. USER INTERFACE
2. DATA EXTRACTION
3. DATABASE
4. DATA COLLECTION

1- USER INTERFACE:

L’utente si relaziona al Sistema attraverso questa componente la quale è fornita di una interfaccia grafica (GUI) tramite cui un utente ha accesso ai dati richiesti già calcolati dal Data Analyzer.

2- DATA EXTRACTION

La componente che si occupa della parte computazionale è il Data Analyzer. Il Data Analyzer si occupa dell’elaborazione dei dati acquisiti dal database per fornire I valori di IR by Robot e IR by Cluster che poi verranno visualizzati sulla dashboard.

3- DATBASE

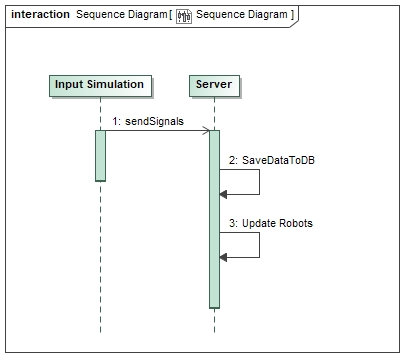
Il Database riceve i dati da memorizzare dalla componente DATA COLLECTION e fornisce I dati alle richieste del Data Analyzer.

4- DATA COLLECTION

È la componente che si occupa dello storage sul database dei dati ottenuti in ingresso dai robot.

C.2 The dynamic view of the software architecture: Sequence Diagram

*BE SURE THAT THE STRUCTURE IS SYNCHRONIZED WITH THE DYNAMIC VIEW*



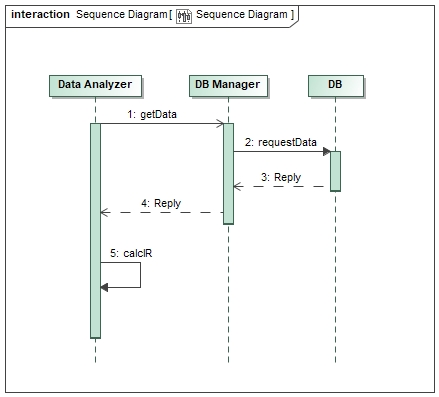
**Figura 4- Sequence Diagram Scenario#1 (Data Storage)**

Scenario che illustra il processo di invio dei segnali da parte dei robot ed il salvataggio degli stessi.

1 – sendSignal: Invio dei segnali del robot

2 – saveData: Trasferimento dei dati al controller

3 – Data Storage: Componente che effettua il salvataggio dei dati ricevuti sul database.



**Figura 5- Sequence Diagram Scenario#2 (IR Calc)**

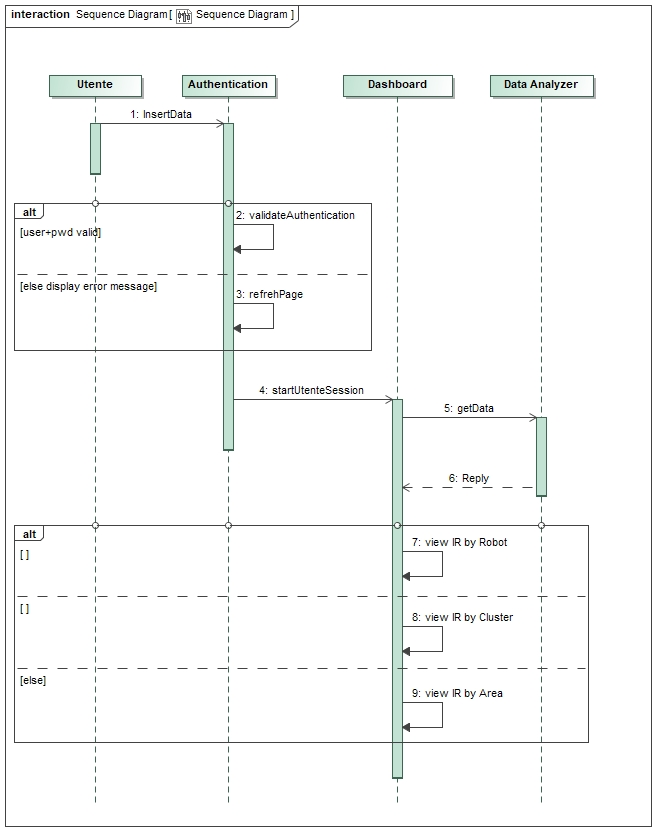
Scenario che mostra il processo di data extraction, ovvero l’analisi dei dati sul database e produzione della percentuale di inefficiency rate.

1 – getData: dataAnalyzer recupera I dati dal database

2 – reply: Il database risponde alla richiesta del data analyzer passandogli le informazioni richieste.

3 – calcIR: Effettua il calcolo dell’IR

4 – calculatedIR - Restituisce l’IR calcolato.



**Figura 6- Sequence Diagram Scenario#3 (User)**

Scenario che mostra l’interazione dell’utente con la dashboard e la visualizzazione dei dati.

1 – insertData: L’utente inserisce I dati necessari per l’autenticazione

2 – validateAuthentication: (if) Controllo di validità sui dati inseriti per completare l’autenticazione.

3 – refreshPage: (else) Refresh della pagina per nuovo tentativo

4 – startUtenteSession: Avvio della sessione utente

5 – getData (da rinominare): La dashboard effettua una richiesta per il recupero dei dati di IR.

6 - reply: Invio dei dati alla dashboard.

7 – requestIRVisualization: L’utente invia una richiesta per visulizzare le informazioni desiderate (IR by robot o by cluster).

8 - view IR by cluster: (if) imposta la visualizzazione dell’IR by cluster

9 - view IR by robot: (else) imposta la visualizzazione dell’IR by robot

10 – replyToDashboard: La dashboard mostra le informazioni desiderate dall’utente.

D. ER Design

<Report here the Entity Relationship Diagram of the system DB>

E. Class Diagram of the implemented System

F. Design Decisions   
<Document here the **5** most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

THIS IS A VERY IMPORTANT PART. BE SURE TO DOCUMENT THE 5 MOST IMPORTANT DECISIONS (related to your requirements and design) YOU MADE

G. Explain how the FRs and the NFRs are satisfied by design

<Report in this section how the design you produced satisfies the FRs and the NFRs>

H. Effort Recording

***GANTT****Make a GANTT documenting the tasks and timing you expect to spend on the deliverable. Try to be as precise as possible. Check, after the deliverable deadline, if and how you satisfied (or not) the deadlines.*

***Logging*** *As you are working on the assignment, record what you are doing and how long you spent. As a rule of thumb, you should add a log entry every time you switch tasks. For example, if you do something for two hours straight, that can be one log entry. However, if you do two or three things in half an hour, you must have a log entry for each of them. You do not need to include time for logging, but should include the time spent answering the other parts of this question.*

*For this purpose, please use the* ***LogTemplate.xls*** *file.*

***Categorization*** *When logging the time spent on the project, please create different sub- categories.**Specifically, it is important to clearly distinguish between two main categories: the time spent for “****learning****” (the modeling languages, the tools, etc.) from the time needed for “****doing****” (creating the models, taking the decisions, …). Learning tasks are in fact costs to be paid only once, while doing costs are those that will be repeated through the project.*

*For each category, please define sub-categories. Examples follow. You may add other sub-categories you find useful.*

|  |  |
| --- | --- |
| ***Learning***   * ***Requirements Engineering*** * ***Non functional Requirements*** * ***Use Case Diagrams*** * ***Tool study*** | ***Doing:***   * ***Requirements discovery*** * ***Requirements Modeling (UC diagrams)*** |

***Summary Statistics****Based on the attributes defined above, calculate the summary statistics of the time spent for “learning”, the time spent for “doing”, and the total time.*

***Note: this Deliverable report shall document only the Summary Statistics for the different deliverables (D1, D2, and Final). Detailed information shall be reported in the Excel file.***

***COPY HERE (computed from the spreadsheet): i) the total number of hours spent by the group (that is, hours per task X number of people working on that task), ii) the time spent for LEARNING and for DOING***

Appendix. Code   
<Report in this section a **documented** version of the produced code>  
<Discuss how your code satisfies the FR and NFR>  
<Show some screenshots of the code behavior>