

# **SmartForklift**

Pervasive computing

Simone Del Gatto - 952242 simone.delgatto@studio.unibo.it  
Sophia Fantoni - 933149 sophia.fantoni@studio.unibo.it

April 2021

# Indice

<b>1 Analisi del dominio</b>	<b>4</b>
1.1 Organizzazione del magazzino . . . . .	4
1.2 Organizzazione attuale del lavoro . . . . .	6
1.2.1 Problematiche . . . . .	7
1.2.2 Obiettivi . . . . .	7
1.3 Casi d'uso . . . . .	8
<b>2 Requisiti</b>	<b>10</b>
2.1 Requisiti di business . . . . .	10
2.2 Requisiti utente . . . . .	10
2.3 Requisiti funzionali . . . . .	11
2.4 Requisiti di implementazione . . . . .	11
<b>3 Design</b>	<b>12</b>
3.1 Elementi del sistema . . . . .	12
3.2 Scenario di utilizzo . . . . .	14
<b>4 Progettazione e sviluppo</b>	<b>17</b>
4.1 IoTServer . . . . .	20
4.2 SmartForklift device . . . . .	22
4.3 Multiorders application . . . . .	25
4.3.1 Dispositivi hardware . . . . .	25
4.3.2 Struttura dell'applicazione . . . . .	26
<b>5 Considerazioni finali</b>	<b>29</b>
<b>6 Validazione</b>	<b>30</b>
6.1 Demo . . . . .	30

6.2 Guida utente . . . . .	30
----------------------------	----

# Introduzione

Il progetto *SmartForklift* nasce in collaborazione con Life365 s.p.a, azienda forlivese che si occupa di commercio B2B di prodotti elettronici, ed ha come obiettivo quello di migliorare la produttività e la sicurezza degli operai all'interno del magazzino.

A tale scopo verranno utilizzate alcune tecnologie come:

- *Smart glasses* (Vuzix Blade);
- *Lettore bar code* (Netum);
- *Raspberry e attuatori*.

Nel primo capitolo viene fatta l'analisi del dominio, al fine di comprendere a fondo i concetti, le dinamiche, le caratteristiche e il funzionamento del magazzino dell'azienda. Successivamente vengono descritti i principali requisiti del sistema. Nel capitolo tre e quattro osserveremo il design del sistema, oltre alla parte di progettazione e sviluppo. Infine, nell'ultimo capitolo viene presentata la validazione del sistema, quali saranno i possibili sviluppi futuri e una breve guida utente.

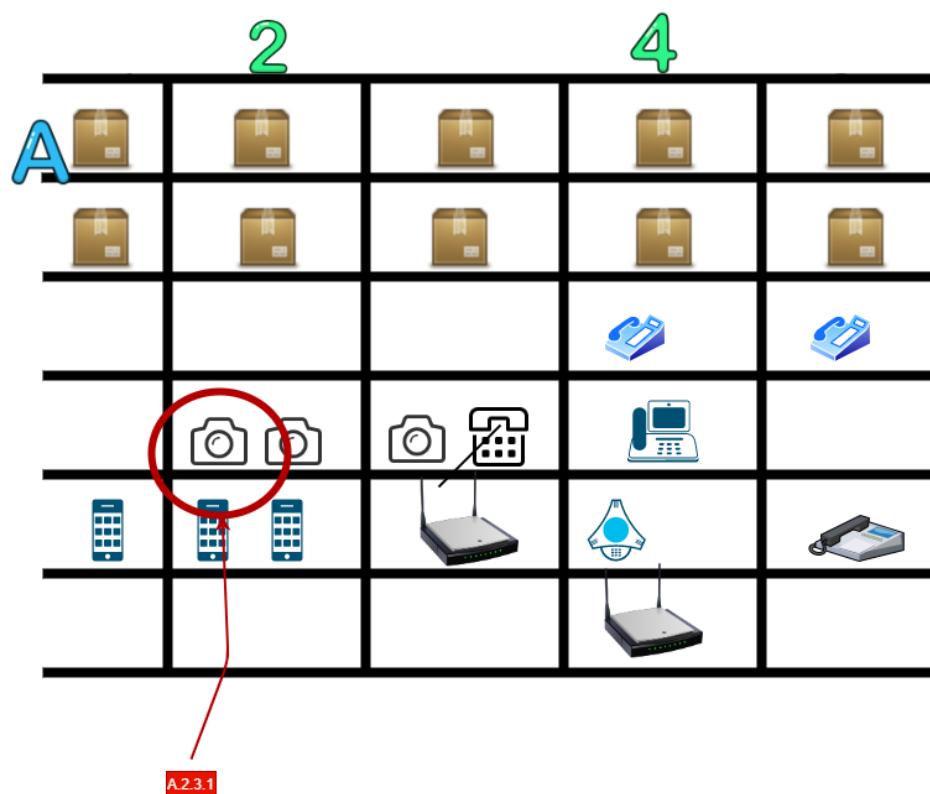
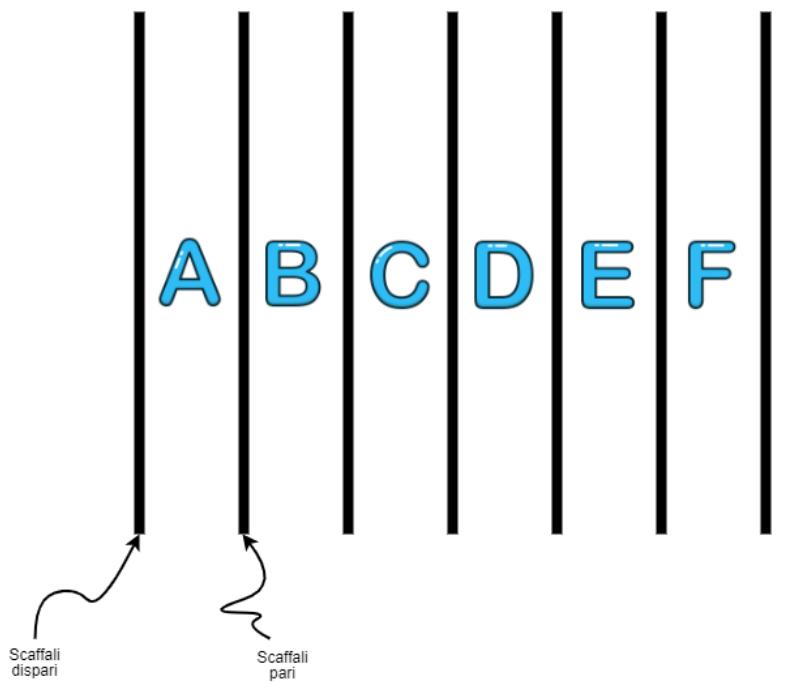
# Capitolo 1

## Analisi del dominio

*Life365 Italy S.p.A.* è un'azienda forlivese che si occupa di commercio B2B di prodotti elettronici. Si presenta sul mercato italiano come una società ponte tra Europa ed Asia, con contratti di import & export con alcuni tra i più importanti produttori di Cina, Giappone, Korea, Taiwan, Honkong e Singapore, per poi distribuire la merce su tutto il territorio nazionale. Per le sue attività Life365 dispone di un ampio magazzino di **tot** mq<sup>2</sup> in cui sono stoccati circa 18.000 prodotti e in cui lavorano quindici magazzinieri.

### 1.1 Organizzazione del magazzino

I prodotti all'interno del magazzino sono disposti su 18 file di scaffali, enumerate dalla A alla R, ogni scaffale contiene dai 3 ai 5 piani, solo un paio sono considerati posizioni *a terra* (cioè accessibili dai magazzinieri, quindi dalle quali essi possono prelevare i prodotti), i rimanenti vengono utilizzati per il deposito delle merci.



Per comprendere al meglio l'organizzazione del magazzino osserviamo la *figura 1.1*. Ogni posizione all'interno del magazzino è formattata come stringa del tipo “A.2.3.1”:

- la lettera *A*: indica la fila di scaffali;
- il primo numero *2*: indica lo scaffale;
- il secondo numero *3*: indica il piano;
- l'ultimo numero *1*: indica la posizione all'interno del piano.

Facendo un'analogia con il sistema urbano, le lettere corrispondono ai nomi delle vie, inoltre, gli scaffali sono enumerati mantenendo da un lato tutti i numeri dispari e dall'altro quelli pari in maniera progressiva.

## 1.2 Organizzazione attuale del lavoro

I magazzinieri sono professionisti che, assieme ad altri collaboratori, hanno il compito di gestire il magazzino e tutto ciò che vi ruota attorno, occupandosi quindi della ricezione, dello stoccaggio, della preparazione e quindi della spedizione delle merci. L'attività di interesse per l'applicazione è chiamata **picking** che rappresenta la preparazione degli ordini.

Attualmente ogni magazziniere dispone di:

- un muletto elettronico, che permette un agile e veloce spostamento all'interno del magazzino;
- palmare (picker) con lettore di barcode, che guida il magazziniere in tutta la procedura.

La lavorazione degli ordini attuale prevede i seguenti step:

1. L'ufficio ordini, che si occupa delle relazioni con i clienti, emette una *bolla* cartacea in cui sono riportate le informazioni principali dell'ordine, in particolare l'identificativo e il suo barcode.
2. Attraverso il palmare il magazziniere inserisce l'id dell'ordine, questa operazione notifica ai sistemi di Life365 l'intenzione di iniziare la preparazione dell'ordine. Avvenuto ciò, il sistema invia all'utente tutte le informazioni utili alla preparazione dell'ordine: tra cui tutti i prodotti e le relative posizioni all'interno del magazzino.

3. Il magazziniere, utilizzando il muletto, raccoglie i vari prodotti e compone l'ordine sulla pedana del muletto. Per ogni prodotto raccolto l'operatore, utilizza lo scanner di barcode del palmare per scansionare il barcode del prodotto. Il picker quindi comunica ai sistemi aziendali l'acquisizione del prodotto, in questo modo si tiene traccia tutte le operazioni degli operatori all'interno del magazzino.
4. Il magazziniere deposita il prodotto nella quantità segnalata dal palmare sulla pedana del magazzino e dà conferma sul dispositivo.
5. Raccolti tutti i prodotti, il picker segnalerà al magazziniere il completamento dell'ordine, egli quindi, depositerà l'ordine per poi prenderne un altro in carico.

### 1.2.1 Problematiche

Nell'attuale procedura di preparazione degli ordini sono presenti alcune problematiche. Innanzitutto, la procedura standard non offre alcun supporto per la preparazione di più ordini contemporaneamente. Questo aspetto non è da sottovalutare in quanto porterebbe ad un notevole aumento della produttività a parità di personale e attrezzatura.

Inoltre, il picker attualmente utilizzato dai magazzinieri li obbliga ad avere, durante il lavoro e per la maggior parte del tempo, le mani impegnate. Ciò risulta non solo scomodo nel momento in cui deve trasportare un prodotto di grandi dimensioni (ad esempio un televisore), ma poco sicuro in quanto spinge l'operaio a guidare il muletto in maniera scorretta e pericolosa con il palmare in mano, togliendo lo sguardo dalle corsie per osservare lo schermo per individuare le posizioni dei prodotti.

### 1.2.2 Obiettivi

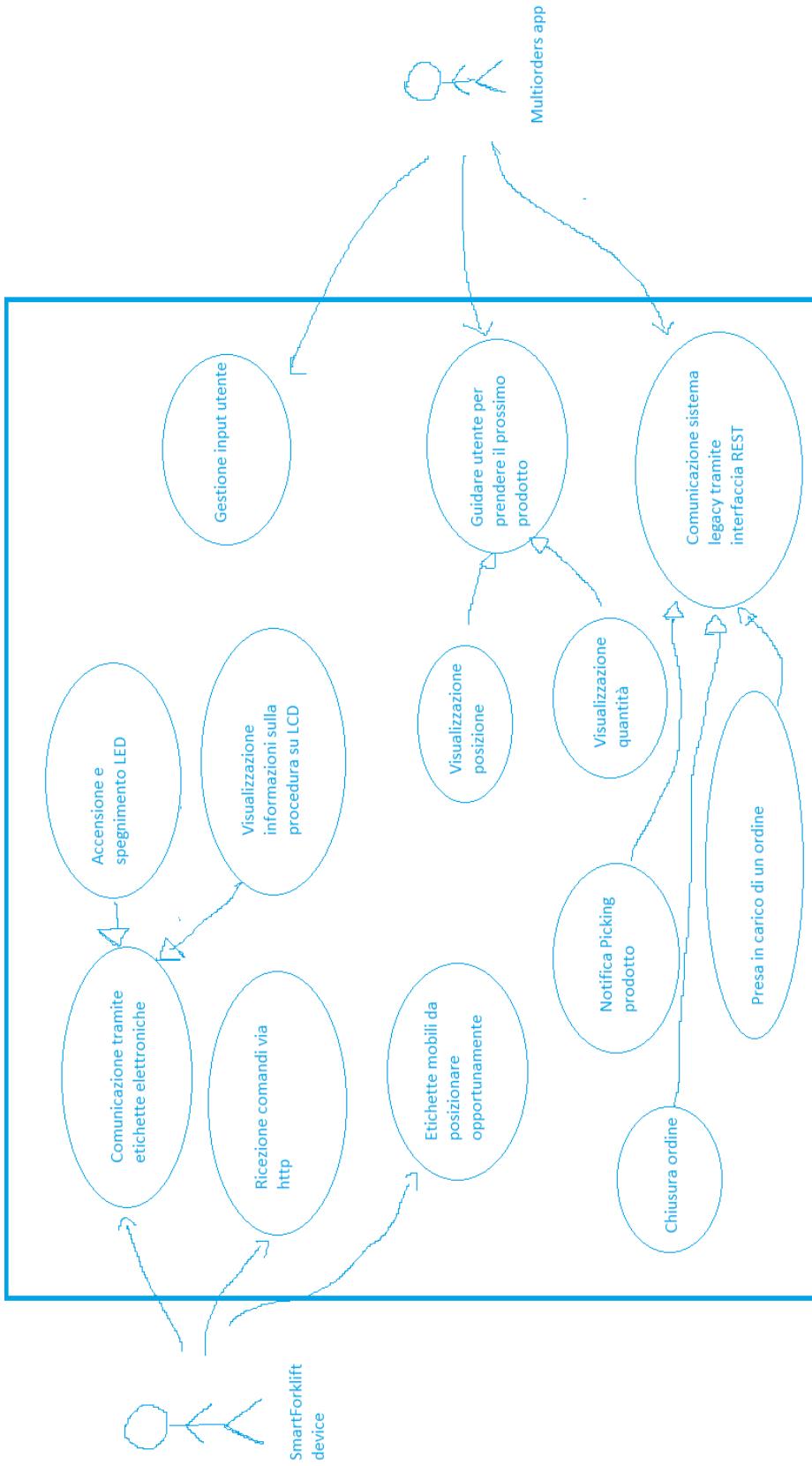
L'elaborato di progetto ha come scopo la realizzazione del sistema SmartForklift che permetterà di:

- migliorare la **produttività** del magazzino intervenendo nell'attività di picking degli ordini, in particolare fornendo il supporto necessario per la preparazione di più ordini contemporaneamente;
- rendere più **agevole** tutta l'attività lavorativa facendo sì che il magazziniere possa lavorare con le mani libere;

- aumentare la **sicurezza** di tutta la procedura fornendo il supporto necessario per non portare il magazziniere ad assumere comportamenti che potrebbero risultare rischiosi per la sua incolumità e quella degli altri (arrampicarsi sugli scaffali con le mani occupate, guidare guardando il palmare);
- continuare ad aggiornare il sistema informativo di Life365, come nelle tecnologie preesistenti, per comunicare la presa in carico degli ordini, il prelevamento dei prodotti e la chiusura della lavorazione di un prodotto.

### 1.3 Casi d'uso

A seguito della fase di analisi osserviamo il diagramma dei casi d'uso:



# Capitolo 2

## Requisiti

In questa sezione verranno trattati in modo dettagliato i requisiti dell'applicazione realizzata.

### 2.1 Requisiti di business

I requisiti di business previsti dal sistema sono:

1. realizzare un sistema che si interfacci a quello già esistente;
2. dare la possibilità di gestire più ordini contemporaneamente;
3. avere una struttura semplice da adattare in base al numero e agli ordini da soddisfare;
4. permettere di visualizzare alcune informazioni degli ordini in gestione, come codice prodotto e quantità;
5. osservare la posizione di ogni prodotto in modo chiaro;
6. differenziare in modo semplice quali prodotti appartengono ad un determinato ordine.

### 2.2 Requisiti utente

Chi farà uso dell'applicazione dovrà quindi essere in grado di:

1. Associare più ordini contemporaneamente.
2. Soddisfare gli ordini in modo semplice, visualizzando le informazioni come:
  - 2.1. codice prodotto,
  - 2.2. quantità,
  - 2.3. ordine di appartenenza,
  - 2.4. posizione all'interno del magazzino.
3. Muoversi nell'ambiente in maggior sicurezza:
  - 3.1. grazie all'introduzione degli smart glasses,
  - 3.2. mani libere durante il lavoro.

## 2.3 Requisiti funzionali

I requisiti funzionali di cui il sistema dovrà disporre sono:

1. Possibilità di gestione più ordini
  - 1.1. senza obbligo di un numero minimo.
2. Interazione con il sistema legacy:
  - 2.1. per l'acquisizione delle informazioni relative agli ordini;
  - 2.2. per ottenere e visualizzare i dati dei prodotti;
  - 2.3. per notificare la presa in carico degli ordini;
  - 2.4. per aggiornare lo stato di lavorazione di un ordine;
  - 2.5. per comunicare la chiusura di un ordine.

## 2.4 Requisiti di implementazione

I requisiti di implementazione che il simulatore dovrà garantire sono:

1. strutturazione del software in modo da poter modificare le informazioni visualizzabili in modo semplice;
2. strutturazione del software in modo da adattarsi semplicemente a possibili cambiamenti nelle risposte delle chiamate al sistema.

# Capitolo 3

## Design

### 3.1 Elementi del sistema

Dopo l'analisi del dominio e l'approfondimento dei requisiti, abbiamo osservato alcuni elementi che risultano essenziali all'interno del sistema (che osserviamo nelle *figura 3.1*):

- **Dispositivo IoT** che estenda le funzionalità del classico muletto,
- **Applicazione android** installata sugli smart glasses per l'interazione dei magazzinieri con il sistema,
- **Lettore barcode** per semplificare e sveltire l'acquisizione degli identificativi degli ordini e dei prodotti.

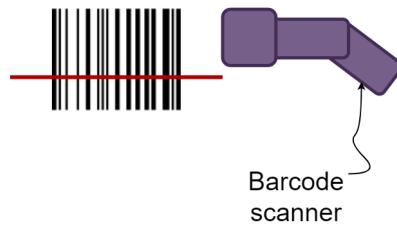
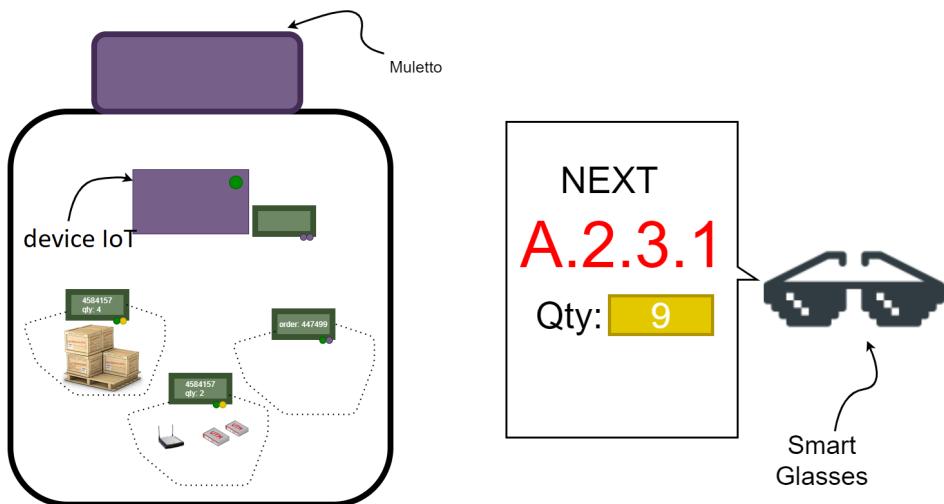


Figura 3.1: Elementi principali del sistema

- **Device IoT**, dispositivo che accresce le funzionalità del classico muletto fornendo il supporto per la preparazione di più ordini contemporaneamente. Si tratta di un device da porre sulla pedana del muletto, composto da varie etichette elettroniche (realizzate con schermo LCD e due led) che, una volta disposte sulla pedana divideranno la base del muletto in zone, ognuna di esse sarà associata ad uno specifico ordine e a tutte le informazioni relative. Sarà possibile comunicare con il dispositivo tramite protocollo http, al fine di effettuare una serie di azioni predefinite.

Ogni *Device IoT* sarà corredata da un opportuno barcode che lo iden-

tifica, tramite questo sarà possibile conoscere il numero massimo di ordini gestibili dal sistema, oltre ad avere il riferimento per l'invio delle richieste.

Anche ogni *etichetta* disporrà di un codice a barre identificativo. Questo sarà essenziale per il mapping tra etichette e ordini, permetterà quindi differenziare i prodotti relativi ad ordini diversi e mostrarne la relativa quantità.

- **Applicazione android** che verrà installata sugli smart glasses. L'applicazione permetterà di visualizzare le stesse informazioni che in precedenza erano fornite dal palmare a disposizione dei magazzinieri. In particolare: le posizioni dei prodotti che compongono gli ordini (una per volta), la quantità richiesta relativa ad ogni ordine oltre al codice prodotto per dare maggior sicurezza all'operatore. L'applicazione si interfacerà con il *device IoT* impartendogli i giusti comandi e con il sistema legacy dell'azienda per il tracking delle attività del magazzino.
- **Lettore barcode** che comunicherà via bluetooth con gli occhiali per la scansione degli identificativi degli ordini e codice dei prodotti. Permette di acquisire i barcode dei prodotti/ordini in modo semplice, rapido e meno soggetto ad errori.

## 3.2 Scenario di utilizzo

Ogni magazziniere sarà dotato del proprio *device IoT* posto sul suo muletto, dal suo personale paio di occhiali intelligenti (con l'*applicazione andorid* installata) oltre al *lettore di barcode*.

La nuova procedura di picking sarà la seguente:

1. Il magazziniere indosserà gli occhiali, collegherà il lettore barcode tramite Bluetooth e farà partire l'applicazione.
2. L'operaio indicherà quale *device* vorrà utilizzare scansionando il codice a barre riportato sul dispositivo.
3. Il magazziniere dovrà poi associare ogni ordine ad una specifica etichetta elettronica disponendole sul carrello a suo piacimento e in base alla sua esperienza. Per associare ordine ed etichetta il magazziniere deve utilizzare il barcode dell'ordine presente nella bolla cartacea e il

barcode presente sull'etichetta. Ogni etichetta che viene associata a un ordine segnala l'avvenuta associazione con l'accensione del led blu e visualizzando l'id dell'ordine sullo schermo.

4. Lasciandosi indirizzare dalle informazioni visualizzate con gli occhiali e dagli output delle etichette, l'operatore deve procedere con l'attività di picking. Attraverso gli occhiali apprenderà le informazioni relative al prossimo prodotto da prelevare: la sua posizione e la quantità necessaria.
5. Prelevato il prodotto nella quantità indicata dovrà scansionare il barcode del prodotto (per avere una maggiore certezza e un minor rischio di errore) e porlo nello spazio occupato dall'ordine indicato dalle etichette. Il led verde associato all'ordine del prodotto scansionato, si accenderà e una dicitura che indicherà la quantità di prodotto per quello specifico ordine verrà visualizzata sul LED.
6. Una volta completata l'acquisizione del prodotto (nella quantità indicata), il magazziniere dovrà confermare l'operazione di picking dei prodotti.
7. Dopo aver concluso un ordine il sistema renderà l'etichetta nuovamente disponibile per l'associazione ad un nuovo ordine.
8. Il magazziniere procede così fino alla lavorazione completa di tutti gli ordini, questa verrà notificata al magazziniere attraverso l'interfaccia dell'applicazione.

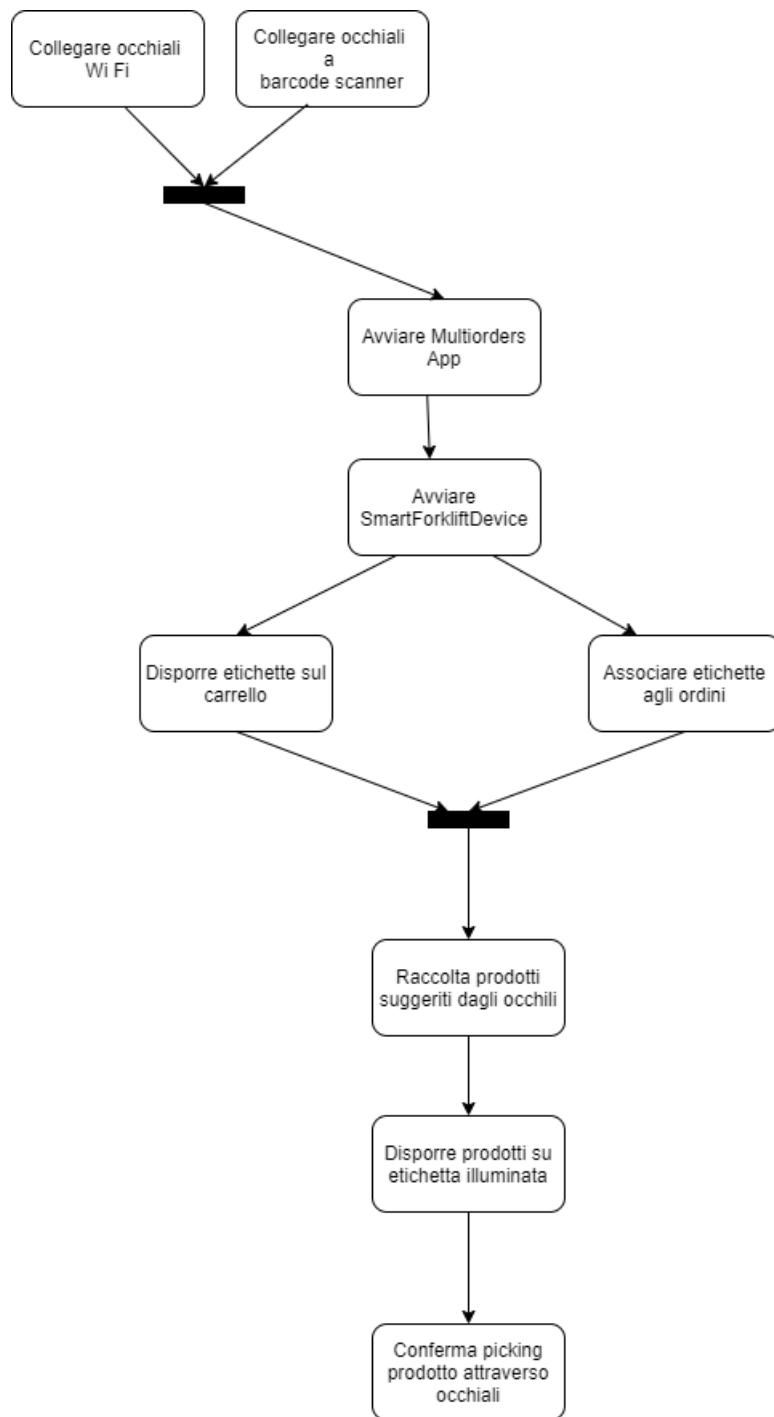


Figura 3.2: Procedura di Picking proposta

# Capitolo 4

## Progettazione e sviluppo

Per la realizzazione del sistema sono stati utilizzati i seguenti dispositivi esterni: Vuzix Blade, occhiali per la realtà aumentata, e Netum, lettore di barcode da connettere tramite bluetooth.

Le due componenti che sono state ben identificate sono:

- **SmartForklift device**, dispositivo che integra i muletti presenti nel sistema dando la possibilità di gestire più ordini;
- **Applicazione Multiorders**, installata sui Vuzix Blade per l'interazione tra sistema e magazzinieri.

Tra le problematiche del sistema vi è quella di rendere visibili i vari SmartForklift device presenti all'interno del magazzino. I dispositivi infatti, hanno una connessione che gli consente di accedere alla rete, ma non hanno la possibilità di mettere a disposizione un indirizzo IP pubblico con cui poter essere contattati. Per risolvere tale difficoltà è stato aggiunto il componente

- **IoTServer**, server intermedio che si occupa di esporre i dispositivi sulla rete, intercettare le chiamate delle applicazioni Multiorders e di ridirezionare le informazioni sugli opportuni device.

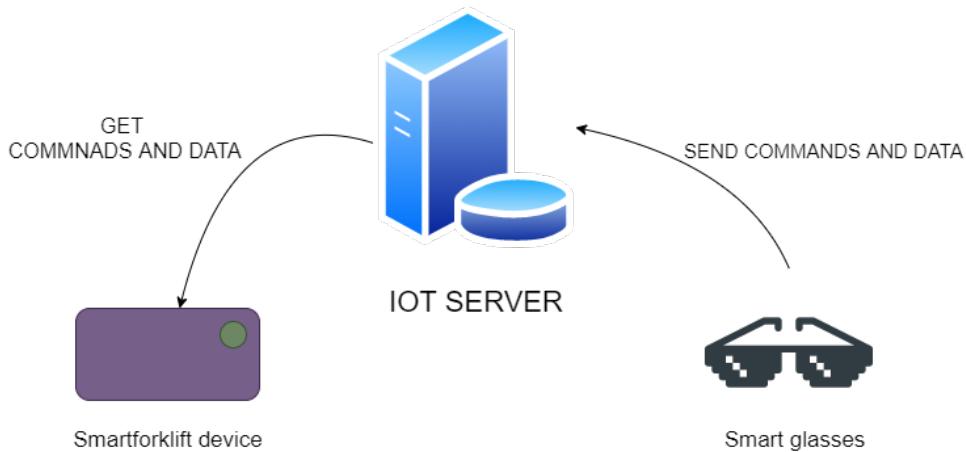


Figura 4.1: Iot Server come intermediate element

Il comportamento delle componenti del sistema e lo svolgimento della procedura di picking sarà il seguente:

- Una volta acceso, lo *SmartForklift device* pubblica sul server IoT una sua *things description* in cui notifica il suo id, quante sono le etichette (*placements*) che gestisce, i loro identificativi, e le azioni che possono essere fatte sul dispositivo.
- Quando viene avviata l'*applicazione Multiorders* questa si aspetta l'inserimento dell'identificativo dello *SmartForklift device* con cui relazionarsi (nell'ottica in cui il sistema ne possa gestire più di uno). Una volta ottenuto richiede al Server Iot gli identificativi dei placement gestiti dal dispositivo e permette all'operatore di associarli ad un ordine.
- Ogni volta che avviene un'associazione l'applicazione comunica con il sistema aziendale per ottenere tutte le informazioni relative all'ordine, in particolare la lista dei prodotti e la quantità relativa ad ognuno.
- L'*applicazione multiorders* mostra quale debba essere il prossimo prodotto da prelevare, in particolare le informazioni relative alla posizione, al codice del prodotto e alla quantità da prelevare.
- Una volta raccolto il prodotto, attraverso il lettore di codice a barre si scansionerà il suo codice del prodotto per indicare il suo prelevamento. *Multiorders* comunicherà le proprie azioni, tramite chiamate

all’*IotServer*. L’etichetta relativa all’ordine di interesse accenderà il suo led verde, in questo modo il magazziniere identificherà in modo rapido l’area su cui dovrà riporre il prodotto.

- Una volta che il magazziniere avrà confermato di aver posizionato il prodotto nello spazio apposito, *Multiorders* procederà a comunicare al sistema legacy l’avvenuta procedura di picking del prodotto, il led verde si spegnerà. Una volta completato un ordine l’etichetta relativa resetterà LCD.
- Una volta completati tutti gli ordini il sistema legacy, l’*applicazione Multiorders* e le etichette verranno notificate.

Si presuppone che all’inizio della procedura per scannerizzare il barcode il lettore Netum sia stato collegato ai Vusix blade tramite Bluetooth.

Per rendere più agevoli le operazioni, si è deciso di far sì che lo *SmartForklift device* presenti fisicamente al suo esterno un’etichetta con codice a barre, che rappresenta il suo codice identificativo all’interno del sistema, utile nel momento in cui *Multiorders* richiede all’utente quali device vuole utilizzare. In questo modo basterà scannerizzare il suo barcode. Anche ogni etichetta del device avrà il proprio specifico barcode in modo da poterli sfruttare nel momento in cui devono essere associati a uno specifico ordine.

Nelle sottosezioni successive si descriveranno più in dettaglio le varie componenti del sistema, come sono state progettate, realizzate e messe a disposizione.

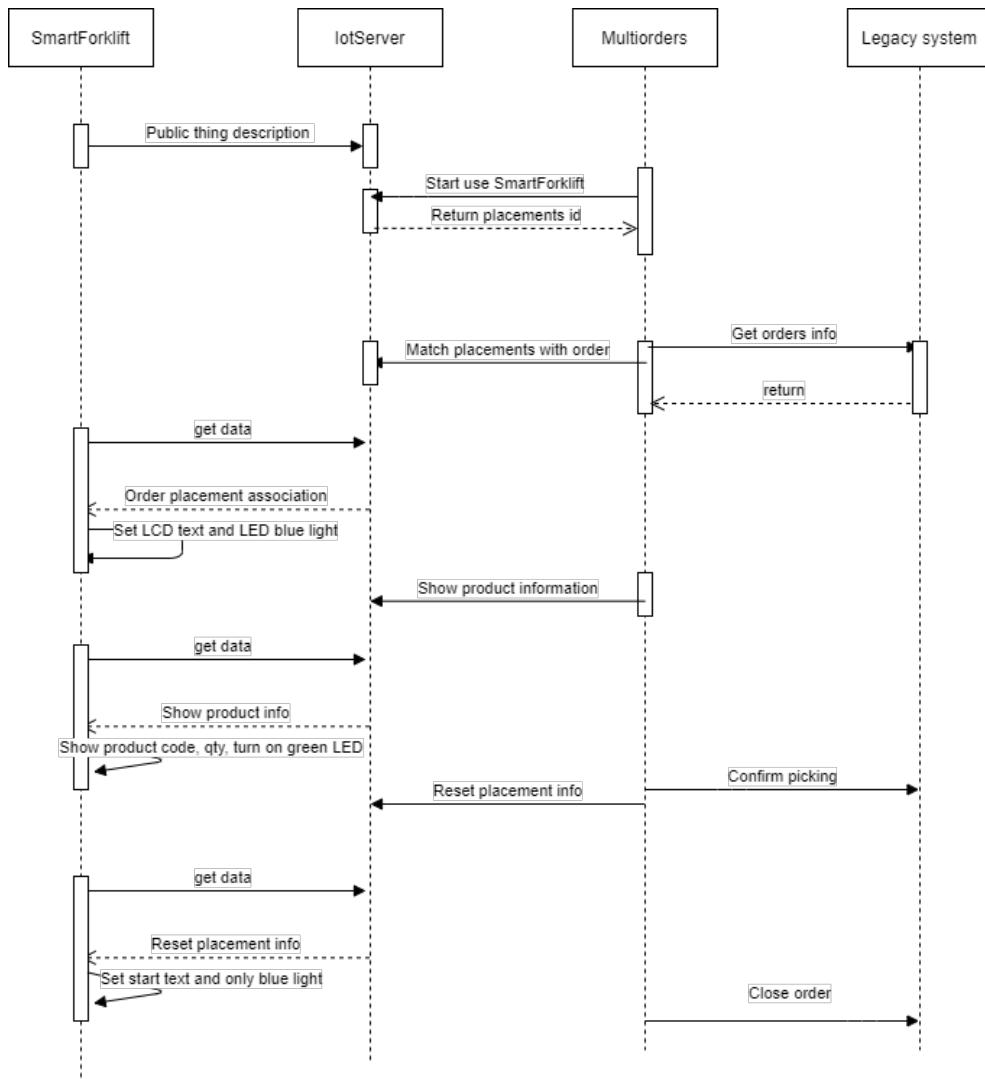


Figura 4.2: Procedura di picking - Diagramma di sequenza

## 4.1 IoTServer

IoTServer è il componente del sistema che ha come compito quello di rendere disponibili le funzionalità dei dispositivi SmartForklift su tutta la rete, permettendone la visibilità in particolare alle applicazioni Multiorders.

Per pubblicizzarsi attraverso IoTServer un dispositivo può inviare una sua descrizione attraverso la chiamata *POST /smartForklift/publicConfig*. Nel body della richiesta dovrà essere incapsulata una descrizione che si ispira alla Thing Description proposta dal W3C.

```
{  
    "placements": [1, 2],  
    "action": ["startUse", "setPlacement", "putItHere"],  
    "id": 1  
}
```

All'interno della descrizione si trovano gli identificativi delle etichette presenti sul device e tutte le azioni che possono essere eseguite sul device stesso e sulle sue etichette.

L'applicazione Multiorders può contattare il servizio attraverso gli endpoint

*POST /smartForklift/deviceId/action/actionName*  
attraverso il quale può richiedere un'azione che il device deve compiere tra quelle previste, e

*POST /smartForklift/deviceId/placements/placementsId/action/actionName*  
attraverso la quale può svolgere un'azione su una specifica etichetta. Tutte le volte che il server riceve una di queste chiamate, esso memorizza l'informazione in una coda di messaggi che può essere letta dal device tramite la chiamata

*GET /smartForklift/deviceId*

La coda di messaggi è stata realizzata utilizzando *Redis* ([link](#)).

Il server è stato realizzato in *Python3.9* utilizzando come libreria *Flask*.

Il deployment è avvenuto su una macchina Ubuntu remota di proprietà di Life365.

Si è deciso di sfruttare la tecnologia di Docker realizzando due container, uno per IoTServer e uno per Redis.

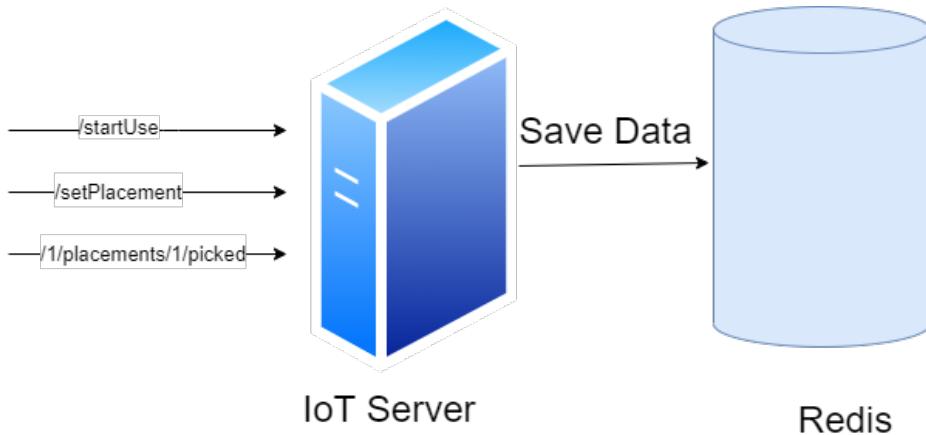


Figura 4.3: Iot Server e coda di messaggi con Redis

## 4.2 SmartForklift device

*SmartForklift device* è il dispositivo IoT che estende le funzionalità del classico muletto fornendo il supporto per la preparazione di più ordini contemporaneamente. Si tratta di un device da porre sulla pedana del muletto composto da varie etichette elettroniche (realizzate con schermo LCD e due led) che, una volta disposte sulla pedana, indicheranno ognuna la posizione spaziale di uno specifico ordine e tutte le sue relative informazioni. Sarà possibile comunicare al dispositivo l'avvio di certe azioni predefinite utilizzando il protocollo http. Ogni device sarà corredata da un opportuno barcode che indica l'identificativo univoco del device. Anche ogni etichetta disporrà di un codice a barre a cui è segnato il suo identificativo utile per l'associazione ordine-etichetta.

Dal punto di vista fisico e hardware il dispositivo è realizzato con i seguenti componenti:

- Raspberry Pi4,
- un modulo I2C multiplexer TCA9548A ,
- due etichette elettroniche composte ognuna da: schermo LCD con interfaccia HD44780, un led blu e uno giallo,
- un led verde,

- una powerbank (5200 mAh, 5V, 2.1A) per l'alimentazione.

La comunicazione tra il Raspberry e gli schermi avviene attraverso protocollo I2C e il multiplexer consente la comunicazione con più schermi contemporaneamente

All'interno del dispositivo è presente un file di configurazione con le seguenti informazioni:

- identificativo del dispositivo,
- una lista in cui ogni elemento rappresenta un placement con le relative informazioni. In particolare:
  - identificativo del placement/etichetta,
  - GPIO del led verde utile per segnalare che l'etichetta è stata associata a un ordine,
  - GPIO del led giallo che indica al magazziniere che un prodotto deve essere posizionato in quel placement,
  - indirizzo I2C dello schermo LCD.
- indirizzo di IoTServer,
- lista di azione che possono essere richieste sul dispositivo,
- intervallo con cui vengono richiesti i messaggi destinati al device a IoTServer.

```
{
  "ID": 1,
  "HARDWARE_SETTINGS": {
    "ready_led_pin": 24,
    "error_pin": -1,
    "placements": [
      {
        "id": 1,
        "ready_led_pin": 15,
        "catch_attention_led_pin": 14,
        "display_channel": 2
    }
  ]
}
```

```

        },
        {
            "id": 2,
            "ready_led_pin": 23,
            "catch_attention_led_pin": 18,
            "display_channel": 3
        }
    ],
},
"ACTIONS": [startUse, setPlacement, putItHere],
"SERVER_ADDRESS": "http://192.168.43.19:5000",
"INTERVAL": 1000
}

```

Il comportamento di *SmartForklift device* può essere schematizzato in questi punti:

- Al momento dell'accensione viene lanciata l'applicazione che regola tutto il comportamento del dispositivo.
- L'applicazione comunica all'*IoTServer* la propria thing description nelle modalità previste descritte nella sottosezione precedente.
- Il dispositivo periodicamente richiede all'*IoTServer* quali azioni gli sono state richieste di seguire e le esegue in ordine cronologico.

Di seguito le azioni che possono essere eseguite sul dispositivo e il relativo output prodotto:

- *startUse*: indica che qualcuno richiede l'utilizzo del dispositivo. Viene acceso il led verde. Vengono spenti i led di ogni etichetta e su ogni schermo compare la dicitura "*READY-TO-CONNECT*"
- *setPlacement*: indica la volontà di associare a un etichetta un ordine. Viene acceso il led verde dell'etichetta corretta e sullo schermo compare la dicitura "*ORDER id ordine*" e "*PICKING*"
- *putItHere*: indica che un determinato prodotto deve essere posto accanto a una specifica etichetta: viene acceso il led giallo dell'etichetta e sullo schermo compare la dicitura con il codice prodotto e la quantità.

- *picked*: richiesta nel momento in cui un magazzinere conferma di aver posizionato i prodotti correttamente. Viene spento il led giallo e torna sullo schermo la dicitura vista in *setPlacement*
- *orderDone*: richiesta nel momento in cui un ordine viene considerato terminato. Sulla specifica etichetta relativa all'ordine tornerà la dicitura iniziale e tutti i led saranno spenti

L'applicazione in esecuzione sul Raspberry è stata realizzata in Python3.9, sfruttando le librerie messe a disposizione dal sistema operativo Raspian per l'interfacciamento con i GPIO relativi ai led, e la libreria *smbus* per la comunicazione I2C con gli schermi LCD.

## 4.3 Multiorders application

**Multiorders** è un'applicazione Android che permette ai magazzinieri di interfacciarsi in modo agevole con il sistema. Questi saranno quindi in grado di osservare le informazioni principali dei prodotti al fine di gestire in modo chiaro e semplice più ordini contemporaneamente.

In questa sezione andremo ad analizzare i dispositivi hardware e la struttura dell'applicazione.

### 4.3.1 Dispositivi hardware

L'applicazione gira sugli occhiali **Vuzix Blade Upgraded Smart Glasses**, questi rappresentano l'elemento chiave in quanto permettono agli operai di avere le mani libere e quindi di lavorare in maggior sicurezza. Questi sono uno dei primi occhiali intelligenti a realtà aumentata completamente autonomi, grazie all'ottica trasparente Waveguide uniscono le istruzioni digitali alle attività del mondo reale, senza ridurre in modo drastico il campo di vista, evitando occlusioni e rimuovendo distrazioni. Oltre a molte funzionalità e sensori che mettono a disposizione, gli Smart Glasses Vuzix Blade Upgrade hanno connessione Wi-fi e Bluetooth, oltre a un touch pad per una facile navigazione.



Figura 4.4: Vuzix Blade Upgraded

Il lettore di barcode NETUM viene connesso tramite Bluetooth agli smart glasses. Questo dispositivo viene “visto” come tastiera bluetooth. Tutte le impostazioni del dispositivo come lingua della tastiera, scan mode, offline mode, possono essere settate tramite scansione di bar code.



Figura 4.5: Vuzix Blade Upgraded

#### 4.3.2 Struttura dell'applicazione

L'applicazione Multiorders ha la necessità di inviare richieste HTTP al server, a questo scopo ha bisogno delle autorizzazioni INTERNET e ACCESO\_NETWORK\_STATE, entrambi sono permessi normali, quindi non ri-

chiedono un'esplicita conferma a runtime, ma vengono garantite all'installazione.

L'applicazione si compone di quattro *Activity* (ognuna associata ad un layout) e un *Service*:

- **MainActivity:** activity principale, la prima visibile al momento dell'avviamento. Presenta solo i bottoni per procedere o uscire dall'applicazione.
- **ConnectActivity:** questa activity ha il compito di inviare le richieste di connessione al Raspberry (identificativo da scannerizzare) oltre a settare le informazioni da mantenere relative al muletto.
- **OrderActivity:** activity che permette di associare un'etichetta ad un ordine, l'operatore dovrà scannerizzare i barcode relativi ai due elementi. Sarà possibile associare uno o più ordini in base al numero di etichette disponibili nel raspberry precedentemente connesso.
- **OrderService:** questo service esegue operazioni di lunga durata in background, non fornisce un'interfaccia utente. Il suo compito è di effettuare richieste http per ottenere tutti i prodotti relativi agli ordini selezionati. Prima di passare tali informazioni all'activity successiva ordina i prodotti per ottimizzare gli spostamenti dei magazzinieri.
- **ShowLocationActivity:** quest'ultima activity permette la visualizzazione della posizione prodotti all'interno del magazzino, oltre a indicare la quantità da prelevare. Dopo aver prelevato il prodotto il magazziniere scannerizzerà il codice per accertarsi che il prodotto sia quello corretto, lo inserirà nella scatola relativa all'ordine e setterà il prodotto picked inviato una POST al sistema.

Per permettere alle activity (e service) di scambiarsi informazioni sono stati creati quattro classi che implementano l'interfaccia **Parcelable**. Questa permette di serializzare i dati, essa descrive una modalità di registrazione di un oggetto con tutti i suoi dati primitivi o un qualsiasi oggetto che a sua volta implementa Parcelable. Le quattro classi appartengono al package `model`:

- **Forklift:** rappresenta il muletto, contiene l'id del raspberry, la mappa che mette in relazione gli ordini con i placement, il numero massimo di placement che possono essere gestiti. Inoltre, mantiene anche il jwt:

stringa che identifica il magazziniere dopo il login nel sistema, questo identificativo viene richiesto in alcune chiamate http.

- **ProductInfo**: contiene le informazioni di un prodotto relative ad un particolare ordine: identificativo (l'unico che non varia in ordini diversi), quantità e un valore booleano che indica se il prodotto è stato inserito in un determinato ordine oppure no.
- **Product**: rappresenta le informazioni del prodotto, come: barcode, codice sbarco, posizione all'interno del magazzino e le relative **ProductInfo**.
- **Order**: è semplicemente la lista di prodotti di tutti gli ordini che vengono gestiti da un magazziniere. Questa classe è necessaria in quanto più activity hanno la necessità di ottenere queste informazioni e l'unico modo per passare elementi da una all'altra è avere classi che implementano **Parcelable**.

Il package **api** contiene le classi che permettono di ottenere gli url delle API per interfacciarsi con l'IotServer (**OrderAPI**) e con SmartForklift device (**RaspberryAPI**).

Infine, per la gestione delle richieste HTTP è stata utilizzata la libreria **Volley**. Questa semplifica e velocizza l'accesso alla rete, inoltre permette una gestione autonoma delle richieste e delle connessioni multiple, fa caching delle risposte sia in memoria che su disco, gestisce le priorità, ha strumenti di log, debug e tracciamento delle attività oltre ad avere varie classi per il supporto dei tipi di richieste più comuni. La classe **Controller** permette di gestire la coda di richieste. Ogni activity/service che dovrà effettuare delle richieste aggiungerà alla coda una **Request**, sarà poi Volley ad eseguirla appena possibile, secondo le sue politiche e condizioni del sistema. In ogni **Request** saranno presenti i riferimenti a due listener: **Response.Listener**, invocato quando la richiesta viene svolta con successo, e **ErrorListener**, in caso di errori. I tipi di richiesta utilizzato nell'applicazione sono: **StringRequest**, **JsonObjectRequest**. Inoltre, ogni **Request** prende il metodo della chiamata HTTP: GET, POST, PUT, DELETE.

# Capitolo 5

## Considerazioni finali

Nel complesso ci si ritiene soddisfatti del lavoro svolto. Nonostante i grandissimi passi che si stanno compiendo nelle tecnologie, siamo consapevoli che l'utilizzo degli smart glasses all'interno di una realtà aziendale può risultare ancora precoce. La messa a fuoco non è ottimale e la batteria non dura per tutto il giorno.

Nell'azienda *Life 365* si sta quindi analizzando il lavoro svolto. L'obiettivo è quello di abilitare i magazzinieri alla gestione di più ordini contemporaneamente. Gli scenari possibili al momento sono due:

1. Utilizzare Smartforklift device installando Multiorders su smartphone Android;
2. Utilizzare Smarforklif device e sviluppare una nuova applicazione web accessibili sempre da smarphone.

Entrambe le soluzione mirano a temporeggiare in attesa che le tecnologie AR raggiungano uno stadio di maturazione consono e a rispettare la difficoltà di adattamento alle nuove proposte degli utenti finali. La seconda opzione si allinea anche allo stack tecnologico generalmente utilizzato dal team di sviluppo aziendale che basa lo sviluppo di nuove app principalmente sulle queste tecnologie.

# Capitolo 6

## Validazione

### 6.1 Demo

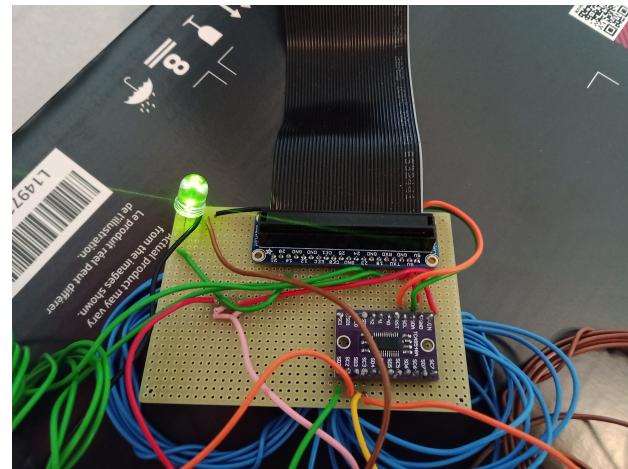
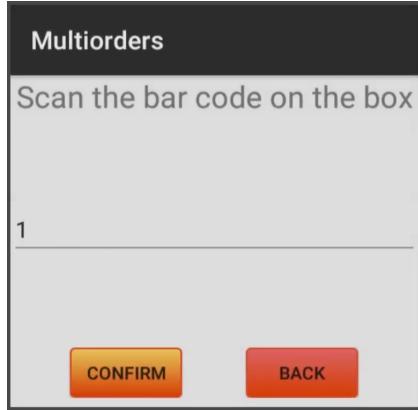
Per validare il sistema è stato realizzato uno SmartForklift device, è stata installata sui Vuzix Blade l'applicazione Multiorders ed è stato effettuato il deployment di IoTServer e del database Redis in modo da consentire la preparazione di due ordini contemporaneamente. Cliccando qui o seguendo l'indirizzo <https://youtu.be/tRpzDQogDoQ> è possibile vedere la demo in cui si effettua la procedura di picking di due ordini.

### 6.2 Guida utente

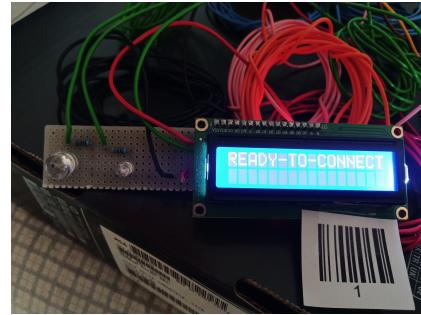
1. *Avvio:* una volta connessi gli occhiali alla rete e al barcode scanner aprire l'applicazione e premere il tasto *CONNECT TO BOX*.



2. **Connessione con SmartForklift device:** Utilizzando Netum, scansionare il barcode che si trova sullo SmartForklift device. Selezionare **CONFIRM**. Il dispositivo risulterà connesso all'accensione del led verde.



Ogni placement mostrerà poi il seguente output:



3. **Associazione ordini-placement:** Scansionare prima il barcode presente sul placement e successivamente il barcode presente sulla bolla dell'ordine. Confermare l'associazione premendo il tasto *CONFIRM*. Se la procedura è andata a buon fine si accenderà il led blu del placement e sullo schermo LCD verrà visualizzato l'identificativo dell'ordine associato all'etichetta.

**Multiorders**

Scan the bar code (ORDER +PLACEMENT)

1

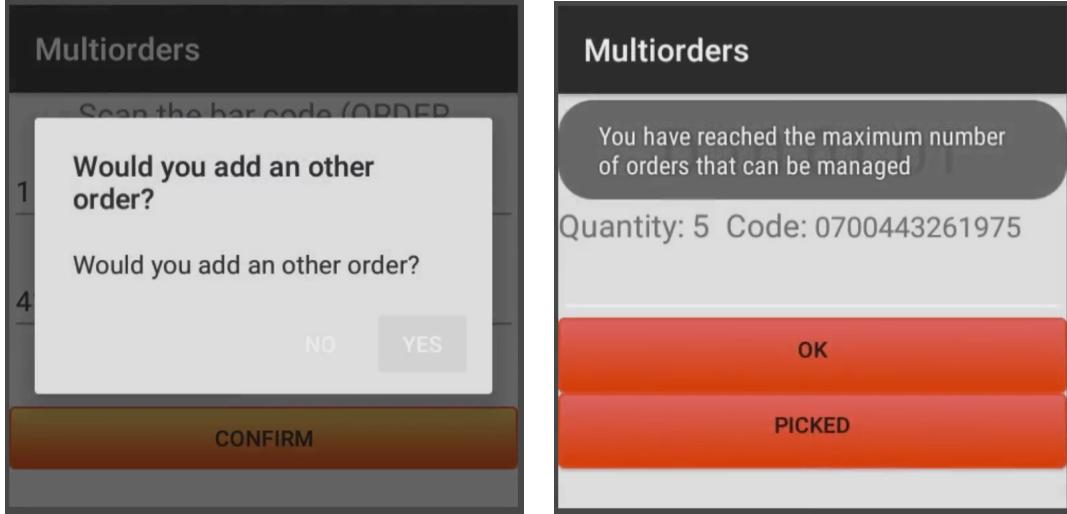
Scan the code of placement

414189

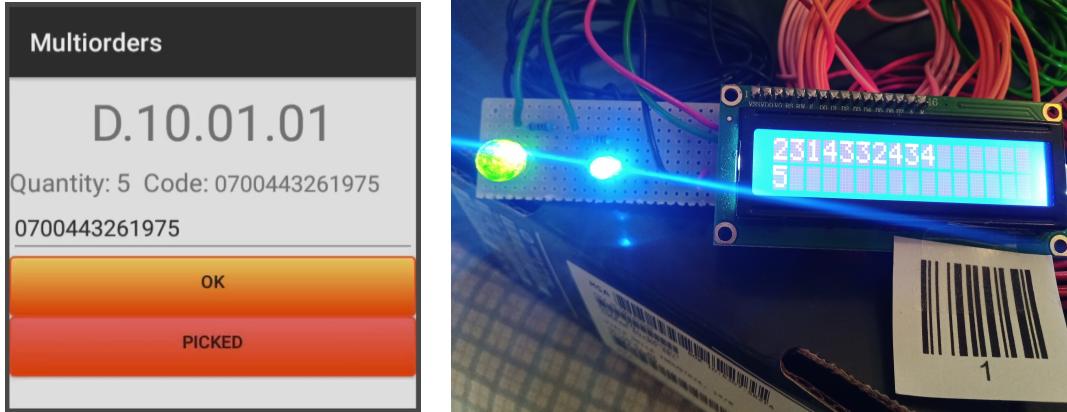
Scan the code of order

**CONFIRM**

L'applicazione chiederà all'utente se vuole effettuare altre associazioni. In caso affermativo premere *YES* ed effettuare la procedura descritta sopra con un'altra etichetta e un altro ordine. La procedura di picking inizierà o alla pressione del tasto *NO* o quando tutti i placement disponibili risulteranno associati. In tal caso l'applicazione mostrerà un messaggio all'utente come da immagine sottostante.



4. **Picking di un prodotto:** dopo aver preso il prodotto corretto, scansionare il barcode e confermare con il tasto *OK*. Si accenderà, quindi, il led verde del placement a cui il prodotto appartiene. Una volta prelevato il prodotto, confermare l'operazione di picking del prodotto (nella relativa quantità) premendo il tasto *PICKED*.



5. **Chiusura ordine:** una volta che sono stati raccolti tutti i prodotti di un ordine, l'etichetta si resetterà e l'output del relativo placement si

mostrera come in figura. Una volta che saranno stati raccolti i prodotti di tutti gli ordini l'applicazione presenterà nuovamente la schermata iniziale notificando il completamento di tutti gli ordini.

