

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



MoviORDER: una piattaforma mobile per la creazione e l'invio di ordini

Tesi di laurea triennale

Relatore

Prof. Armir Bujari

Laureando

Tommaso Carraro

ANNO ACCADEMICO 2017-2018

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

— Oscar Wilde

Dedicato a ...

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dal laureando Pinco Pallino presso l'azienda Azienda S.p.A. Gli obbiettivi da raggiungere erano molteplici.

In primo luogo era richiesto lo sviluppo di ... In secondo luogo era richiesta l'implementazione di un ... Tale framework permette di registrare gli eventi di un controllore programmabile, quali segnali applicati Terzo ed ultimo obbiettivo era l'integrazione ...

“Life is really simple, but we insist on making it complicated”

— Confucius

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. NomeDelProfessore, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, Dicembre 2018

Tommaso Carraro

Indice

1	Introduzione	1
1.1	L'azienda	1
1.1.1	Core business	2
1.2	Offerta di stage	3
1.3	Obiettivi e pianificazione	5
1.4	Rischi	7
1.5	Organizzazione del testo	8
2	Processi e metodologie	9
2.1	Processo sviluppo prodotto	9
3	Background tecnologico	11
3.1	Framework	11
3.1.1	Motivazioni alla base dei framework cross-platform	11
3.1.2	Approcci alla base dei framework cross-platform	13
3.1.3	Xamarin	15
3.1.4	PhoneGap	15
3.1.5	La scelta di PhoneGap	16
3.2	Ambiente di sviluppo	17
3.2.1	Suite di PhoneGap	17
3.2.2	Editor e IDE	18
3.2.3	Gestione DBMS	21
3.2.4	Server web	21
3.2.5	Cloud computing	22
3.2.6	Strumenti di testing	23
3.2.7	Strumenti di versioning e ticketing	23
3.2.8	Strumenti di modellazione e documentazione	24
3.2.9	Linguaggi di programmazione e markup	26
3.2.10	DBMS	26
4	Analisi dei requisiti	27
4.1	Casi d'uso	27
4.1.1	Attori del sistema	28
4.1.2	UC1 - Azioni utente non autenticato	28
4.1.3	UC1.1 - Autenticazione	29
4.1.4	UC1.1.1 - Inserimento username	30
4.1.5	UC1.1.2 - Inserimento password	30
4.1.6	UC1.2 - Visualizzazione errore di autenticazione	30

4.1.7	UC2 - Azioni utente autenticato	31
4.1.8	UC2.1 - Logout	31
4.1.9	UC2.2 - Visualizzazione tutorial applicazione	32
4.1.10	UC2.3 - Gestione carrello	33
4.1.11	UC2.3.1 - Aggiunta articolo	35
4.1.12	UC2.3.1.1 - Inserimento codice articolo	35
4.1.13	UC2.3.1.2 - Inserimento manuale codice articolo	36
4.1.14	UC2.3.1.3 - Inserimento codice articolo con scansione codice a barre	36
4.1.15	UC2.3.1.4 - Inserimento quantità articolo	36
4.1.16	UC2.3.1.5 - Inserimento note articolo	37
4.1.17	UC2.3.2 - Visualizzazione errore di inserimento articolo	37
4.1.18	UC2.3.3 - Modifica articolo	38
4.1.19	UC2.3.3.1 - Modifica quantità articolo	38
4.1.20	UC2.3.3.2 - Modifica note articolo	39
4.1.21	UC2.3.4 - Visualizzazione errore di modifica articolo	39
4.1.22	UC2.3.5 - Selezione singolo articolo	39
4.1.23	UC2.3.6 - Deselezione singolo articolo	39
4.1.24	UC2.3.7 - Selezione totale di articoli	40
4.1.25	UC2.3.8 - Deselezione totale di articoli	40
4.1.26	UC2.3.9 - Rimozione articoli selezionati	40
4.1.27	UC2.3.10 - Invio ordine	41
4.1.28	UC2.3.10.1 - Modifica data ordine proposta	41
4.1.29	UC2.3.10.2 - Inserimento note ordine	42
4.1.30	UC2.3.11 - Visualizzazione errore di mancata selezione di articoli	42
4.2	Requisiti	43
4.2.1	Requisiti funzionali	44
4.2.2	Requisiti qualitativi	54
4.2.3	Requisiti di vincolo	55
4.2.4	Riepilogo requisiti	56
4.2.5	Validazione dei requisiti	56
5	Progettazione e codifica	57
5.1	Tecnologie e strumenti	57
5.2	Ciclo di vita del software	57
5.3	Progettazione	57
5.4	Design Pattern utilizzati	57
5.5	Codifica	57
6	Verifica e validazione	59
7	Conclusioni	61
7.1	Consuntivo finale	61
7.2	Raggiungimento degli obiettivi	61
7.3	Conoscenze acquisite	61
7.4	Valutazione personale	61
A	Convenzioni	63
A.1	Casi d'uso	63
A.2	Requisiti	63

<i>INDICE</i>	xi
B Appendice A	65
Bibliografia	69

Elenco delle figure

1.1	Logo dell'azienda VisioneImpresa	2
1.2	Banner del software VisionENTERPRISE	2
1.3	Logo del marchio movidat	3
1.4	Logo dell'evento Stage-IT	3
1.5	Diagramma di Gantt della pianificazione delle attività di stage	6
3.1	Frammentazione SO del mercato italiano nel 2016	12
3.2	L'obiettivo dei framework cross-platform	12
3.3	Architettura di un'applicazione ibrida	13
3.4	Architettura di un'applicazione interpretata	14
3.5	Architettura del framework Xamarin	15
3.6	Architettura del framework PhoneGap	16
3.7	Figura illustrativa di PhoneGap Build	16
3.8	PhoneGap Desktop App	18
3.9	PhoneGap CLI e PhoneGap App	19
3.10	Logo di Sublime Text 3.0	19
3.11	Logo di Android Studio	20
3.12	Logo di XCode	21
3.13	Logo di Eclipse	21
3.14	Screenshot di SQL Server Management Studio	22
3.15	Logo di Apache Tomcat	22
3.16	Logo di Microsoft Azure	23
3.17	Logo di Postman	24
3.18	Screenshot della console di Google Chrome	24
3.19	Logo di GitHub	25
3.20	Logo di Asana	25
3.21	Logo di Gantt Project e Astah	26
3.22	Logo di TexMaker	26
4.1	Use Case - UC1: Azioni utente non autenticato	28
4.2	Use Case - UC1.1: Autenticazione	29
4.3	Use Case - UC2: Azioni utente autenticato	31
4.4	Use Case - UC2.3: Gestione carrello	33
4.5	Use Case - UC2.3.1: Aggiunta articolo	35
4.6	Use Case - UC2.3.3: Modifica articolo	38
4.7	Use Case - UC2.3.10: Invio ordine	41

Elenco delle tabelle

1.1	Pianificazione oraria del periodo di stage	6
1.2	Analisi dei rischi	7
4.1	Tabella del tracciamento dei requisiti funzionali	53
4.2	Tabella del tracciamento dei requisiti qualitativi	54
4.3	Tabella del tracciamento dei requisiti di vincolo	55
4.4	Riepilogo requisiti	56

Capitolo 1

Introduzione

Al giorno d'oggi la tecnologia sta ricoprendo un ruolo importante nei task di tutti i giorni. In particolare, è indispensabile che un'azienda venditrice di un qualsiasi tipo di prodotti, abbia una propria piattaforma online per la gestione degli ordini. Questo perché le persone sono sempre più abituate ad effettuare ordini online. Infatti acquistare online risulta vantaggioso per molteplici motivi, quali il risparmio di tempo e di denaro e soprattutto la comodità di non doversi muovere da casa per acquistare un prodotto. Per molte aziende non cogliere questo cambiamento potrebbe essere fallimentare, infatti, in futuro, la maggior parte degli acquisti avverrà principalmente online per ogni tipologia di prodotto.

MoviORDER nasce per rispondere a questa necessità, proponendosi come piattaforma universale per la registrazione e l'invio di ordini online. Una qualsiasi azienda interessata a vendere i propri prodotti online può contattare VisioneImpresa per ricevere moviORDER. Dopo un breve periodo di scambio di informazioni riguardanti clienti e prodotti, moviORDER sarà pronta a ricevere ordini online dagli utenti registrati. MoviORDER risulta vantaggiosa per i seguenti motivi:

- * per i clienti:
 - possibilità di trovare i prodotti online e non solo nel negozio fisico;
 - possibilità di risparmiare tempo e denaro dovuto al raggiungimento del negozio fisico;
 - possibilità di controllare in tempo reale la disponibilità dei prodotti.
- * per le aziende:
 - riduce il rischio di fallimento dovuto al cambiamento delle convenzioni degli utenti;
 - aumenta il numero di clienti per l'azienda, permettendo a coloro che non riescono a raggiungere il negozio fisico, di poter comunque effettuare ordini grazie alla piattaforma online.

1.1 L'azienda

VisioneImpresa è un'azienda nuova che da 30 anni si occupa di informatica e più precisamente di quella parte dell'informatica dedicata alle applicazioni gestionali. Inizialmente l'attività di VisioneImpresa era dedicata ad aziende, enti pubblici, studi

professionali e centri di elaborazione dati, gestendo totalmente problematiche informatiche, progettazione di sistemi, hardware, reti, sistemi operativi e software applicativo. Oggi VisioneImpresa punta sulla specializzazione, dedicandosi in modo particolare allo sviluppo del software applicativo e dei relativi servizi di implementazione dello stesso nell'azienda. VisioneImpresa è un team di persone esperte e motivate che opera direttamente su gran parte del Nord Est e indirettamente sull'intero territorio nazionale. VisioneImpresa si rivolge a piccole e medie aziende italiane che intendono impostare sul sistema informatico non solo la semplice gestione amministrativa o di magazzino, ma la completa organizzazione aziendale per affrontare un futuro sempre più complesso e veloce con il supporto di un sistema informatico che aiuti l'azienda a prendere decisioni sempre basate su dati precisi.



Figura 1.1: Logo dell'azienda VisioneImpresa

1.1.1 Core business

VisioneImpresa ha principalmente due core business: VisionENTERPRISE e movidat. VisionENTERPRISE è un software gestionale *ERP_G* dedicato alle medie e piccole aziende industriali, commerciali e dei servizi, che gestiscono notevoli moli di dati e hanno la necessità di lavorare con grande velocità e stabilità. Il software è in grado di collegarsi a tutte le informazioni dell'azienda cliente e di interfacciarsi con tutti i software utilizzati al fine di gestire in modo ottimale l'intera organizzazione aziendale con la massima semplicità e velocità operativa. Il software è in grado di rendere disponibile in tempo reale, alla direzione o al titolare dell'azienda, tutte le informazioni di cui hanno bisogno per prendere decisioni sulla base di dati concreti e oggettivi. Altra qualità di VisionENTERPRISE è la sua completa copertura funzionale: dalla contabilità al magazzino, dall'area commerciale alla produzione, dal controllo di gestione all'E-business.



Figura 1.2: Banner del software VisionENTERPRISE

Movidat è un marchio di VisioneImpresa che progetta e sviluppa applicazioni per dispositivi mobile rivolte alle piccole e medie imprese che vogliono rendere i loro processi più semplici, veloci ed efficienti. Lo slogan di movidat è "ovunque tu sia, il tuo business a portata di mano!", infatti le applicazioni movidat sono rivolte ai dipendenti delle aziende che lavorano in movimento e che devono avere sempre tutto sotto controllo ed essere in grado di agire tempestivamente in caso di problematiche inaspettate. Le soluzioni movidat sono compatibili con la maggior parte dei software gestionali disponibili sul mercato. Tra le soluzioni di miglior successo vi sono:

- * **moviCheck**: applicazione rivolta alle aziende che vogliono offrire alle proprie figure direzionali uno strumento in grado di analizzare in mobilità i dati di business più significativi;
- * **moviSell**: applicazione specificatamente rivolta ai professionisti della vendita, ideata come supporto per la gestione dei rapporti con i clienti.



Figura 1.3: Logo del marchio movidat

MoviORDER rientra tra le applicazioni del marchio movidat andando a soddisfare i bisogni del cliente finale dell'azienda. Infatti, moviORDER permette ai clienti di un'azienda di acquistare i prodotti della stessa direttamente da un'applicazione, installata sugli smartphone degli stessi.

1.2 Offerta di stage

Il 10 Aprile 2018 si è tenuta a Padova la 15esima edizione di Stage-IT, iniziativa che mira ad agevolare l'incontro tra aziende e studenti universitari che puntano ad entrare nel mondo del lavoro con specifico riferimento al settore ICT, favorendo un'occasione di conoscenza reciproca tramite colloqui individuali. Le aziende partecipanti propongono spesso progetti innovativi con ottime opportunità di accrescimento delle competenze individuali. Proprio per questo motivo, lo stagista ha deciso di partecipare attivamente all'evento concludendo positivamente sette colloqui in totale.



Figura 1.4: Logo dell'evento Stage-IT

Le aziende sono state scelte dallo studente per ambito e tecnologie di sviluppo adottate nel progetto. In particolare, lo studente era in cerca di:

- * progetti di sviluppo in ambito mobile (preferibilmente Android): al giorno d'oggi i task si stanno spostando sempre di più dalle postazioni desktop ai dispositivi portatili;

- * progetti di sviluppo di applicazioni web (preferibilmente con utilizzo di framework e librerie Javascript moderne): le skill di programmazione in Javascript sono richieste da gran parte delle aziende che si occupano della realizzazione di applicazioni web.

Tra le varie offerte di stage, VisioneImpresa rientrava in entrambe le preferenze, infatti proponeva un progetto di sviluppo di un'applicazione mobile tramite l'utilizzo di un framework cross-platfom. Nonostante lo stage non prevedesse lo sviluppo in codice nativo Android, permetteva comunque lo sviluppo di un'applicazione mobile. Inoltre, richiedendo l'utilizzo di un framework corss-platform per lo sviluppo era pressoché implicito l'utilizzo di tecnologie web, tra le quali anche Javascript. Per cui, il buon compromesso tra tecnologie conosciute e tecnologie ritenute interessanti ha favorito VisioneImpresa tra le varie offerte analizzate.

1.3 Obiettivi e pianificazione

Il progetto prevedeva la realizzazione di un'applicazione mobile, principalmente Android e iOS, che consentisse ad un utente registrato, tramite la lettura di codici a barre o l'inserimento manuale di codici articolo, di inviare ordini di acquisto al proprio fornitore. L'azienda ha richiesto lo sviluppo delle seguenti funzionalità:

- * **login:** permette ad un utente di essere riconosciuto come cliente dell'azienda. Le credenziali di accesso vengono distribuite dall'azienda insieme all'applicazione;
- * **gestione del carrello:**
 - **aggiunta di un nuovo articolo:** permette all'utente autenticato di aggiungere un nuovo articolo al carrello. L'aggiunta di un nuovo articolo prevede l'inserimento di un codice articolo, tramite scansione di un codice a barre o inserimento manuale e, successivamente, l'inserimento di una quantità da ordinare;
 - **modifica di un articolo:** permette all'utente autenticato di modificare la quantità di un articolo in carrello;
 - **selezione articoli:** permette all'utente autenticato di selezionare uno specifico articolo in carrello oppure tutti gli articoli;
 - **deselezione articoli:** permette all'utente autenticato di deselezionare uno specifico articolo in carrello oppure tutti gli articoli;
 - **rimozione articoli:** permette all'utente autenticato di rimuovere gli articoli selezionati dal carrello;
 - **invio ordine:** permette all'utente autenticato di inviare un ordine composto dagli articoli selezionati in carrello.
- * **invio di e-mail di conferma:** nel caso in cui un ordine venga inviato con successo, il sistema deve inviare all'azienda e all'utente autenticato una mail di conferma contenente un riepilogo dell'ordine.

I prodotti attesi per il termine dello stage erano:

- * **analisi dei requisiti** per l'applicazione da realizzare: partendo dalle specifiche e dalla microanalisi ricevuta, lo stagista doveva definire le funzionalità offerte dall'applicazione, i dettagli dei *web services*_G di comunicazione tra database e front-end, e l'interfaccia grafica dell'applicazione;
- * **applicazione moviORDER:** sviluppo dell'applicazione in ambiente multiplatforma in seguito alla scelta del framework ritenuto più opportuno;
- * **manuali** utente e sviluppatore.

Per adempiere a questi obblighi lo stagista ha pianificato delle attività che sono state concordate con il tutor aziendale. Lo stage prevedeva 320 ore di lavoro che sono state distribuite nel periodo tra l'inizio e la fine dello stage, 2 Luglio 2018 e 7 Settembre 2018 rispettivamente. La pianificazione oraria delle attività all'inizio dello stage è indicata nella seguente tabella.

Attività	Ore
Formazione assistita sui software gestionali di VisioneImpresa e sulle applicazioni analoghe a moviORDER	40
Formazione individuale sui framework cross-platform e scelta del framework ritenuto più opportuno	40
Ridefinizione delle specifiche comprendente delle soluzioni da realizzare e delle metodologie per implementarle	40
Realizzazione dei web services per l'interazione con il database: servizio di autenticazione, servizio di lettura dati, servizio di scrittura dati e servizio di invio e-mail	40
Realizzazione della business logic dell'applicazione	40
Realizzazione delle interfacce grafiche dell'applicazione: login, gestione carrello, aggiunta/modifica articolo e invio ordine	40
Test di scambio dati tra VisionENTERPRISE e moviORDER	40
Documentazione del codice e stesura del manuale utente e sviluppatore	40

Tabella 1.1: Pianificazione oraria del periodo di stage

Viene di seguito riportato un *diagramma di Gantt_G* che illustra come le attività di stage sono state allocate nel tempo di calendario da parte dello stagista. Il diagramma mostra le dipendenze temporali tra le varie attività e il periodo di ferie che l'azienda ha concesso allo stagista, indicato in viola.

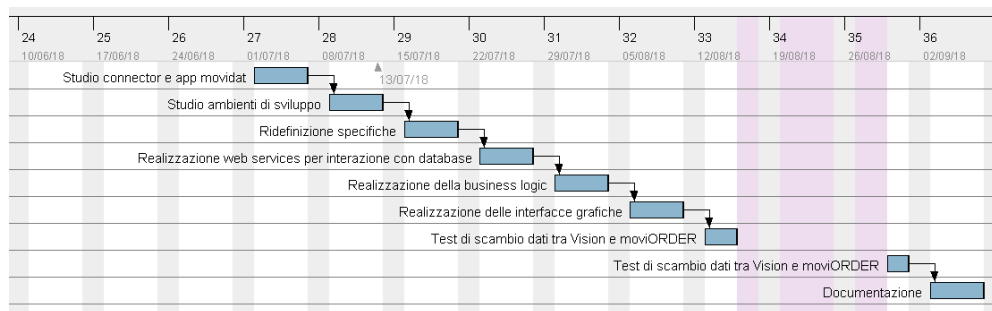


Figura 1.5: Diagramma di Gantt della pianificazione delle attività di stage

1.4 Rischi

Parallelamente allo studio dei documenti forniti dal tutor aziendale, è stata sviluppata un'analisi preventiva dei rischi che sarebbero potuti verificarsi durante lo svolgimento delle attività di progetto. Al fine di attuare una corretta e quindi utile gestione dei rischi essi sono stati identificati nel contesto di processo e di prodotto. In seguito all'identificazione dei rischi sono state analizzate le probabilità di occorrenza e imminenza e l'impatto di ciascun rischio, pianificando come evitarne o mitigarne gli effetti. Durante lo svolgimento del progetto è stato costantemente eseguito un controllo per rilevare eventuali indicatori di rischio e ogni nuovo rischio rilevato è stato aggiunto alla seguente tabella insieme alle azioni atte a mitigarne gli effetti.

Rischio	Descrizione	Grado	Mitigazione
Tecnologie non conosciute	Il tempo richiesto per l'apprendimento delle nuove tecnologie da parte dello stagista potrebbe causare ritardi nello sviluppo	Occorrenza: Media Pericolosità: Media	Effettuare una buona analisi iniziale delle tecnologie richieste ed usufruire del tempo di formazione per prendere dimestichezza con le tecnologie non conosciute
Modifica dei requisiti	Nonostante i requisiti esposti inizialmente siano chiari, vi è la possibilità che questi vengano modificati dal tutor aziendale	Occorrenza: Bassa Pericolosità: Alta	Negoziare la modifica ai requisiti nel caso in cui siano richiesti cambiamenti eccessivi da parte del tutor aziendale
Stima dei tempi	Può accadere che lo stagista abbia pianificato erroneamente alcune attività e che durante lo sviluppo si sforino i tempi concordati	Occorrenza: Media Pericolosità: Alta	Agire tempestivamente sulla pianificazione, ove possibile, nei casi più gravi. Pianificare in maniera intelligente inserendo dei <i>periodi di slack</i> _G tra le attività ritenute più critiche
Difficoltà nelle interazioni	Durante lo stage le interazioni avvengono spesso tramite scambio di e-mail tra lo stagista e il tutor aziendale. Può succedere che alcune risposte arrivino in tempi prolungati causando tempi morti nello sviluppo dell'applicazione	Occorrenza: Bassa Pericolosità: Media	Se il problema dovesse diventare ingestibile proporre al tutor aziendale l'utilizzo di uno strumento di comunicazione alternativo, altrimenti presentarsi fisicamente in ufficio del tutor aziendale per discutere delle problematiche di maggiore entità

Tabella 1.2: Analisi dei rischi

1.5 Organizzazione del testo

Il secondo capitolo descrive ...

Il terzo capitolo approfondisce ...

Il quarto capitolo approfondisce ...

Il quinto capitolo approfondisce ...

Il sesto capitolo approfondisce ...

Nel settimo capitolo descrive ...

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Processi e metodologie

Brevissima introduzione al capitolo

2.1 Processo sviluppo prodotto

Capitolo 3

Background tecnologico

Lo scopo di questo capitolo è la presentazione delle tecnologie utilizzate durante lo sviluppo di moviORDER. La realizzazione dell'applicazione ha permesso l'apprendimento di nuove tecnologie e l'approfondimento di alcune già in parte conosciute. Parte delle tecnologie sono state scelte dallo stagista in seguito al completamento dell'analisi dei requisiti, mentre la maggior parte sono state imposte dal tutor aziendale o dal dominio del problema. Le tecnologie scelte dallo stagista sono state concordate con il team di sviluppo di VisioneImpresa. Le prossime sezioni presentano le tecnologie in base al contesto in cui sono state utilizzate.

3.1 Framework

La presentazione delle tecnologie utilizzate per lo sviluppo di moviORDER inizia dalla scelta del framework, in quanto la scelta del framework ha importato l'utilizzo di parte dei linguaggi di programmazione usati durante il periodo di stage. Per lo sviluppo del progetto è stato imposto l'utilizzo di un framework cross-platform in quanto è stata richiesta la realizzazione di un'applicazione che funzionasse in ambiente Android e iOS. Data la diversità delle tecnologie richieste per lo sviluppo di codice nativo Android e iOS, e la limitata quantità di tempo a disposizione per la realizzazione del progetto, l'utilizzo di un framework cross-platform era migliore soluzione per portare a termine il progetto nei tempi richiesti. Allo stagista era permesso di scegliere tra due *framework cross-platform*_G: Xamarin e PhoneGap.

Vengono di seguito descritti:

1. le motivazioni alla base dei framework cross-platform;
2. gli approcci alla base dei framework cross-platform;
3. il framework Xamarin;
4. il framework PhoneGap;
5. le motivazioni che hanno portato PhoneGap a prevalere su Xamarin.

3.1.1 Motivazioni alla base dei framework cross-platform

Al giorno d'oggi è impensabile realizzare un'applicazione mobile per una sola piattaforma perché il mercato è eccessivamente frammentato. Quindi se si dovesse scegliere di

sviluppare un'applicazione per una sola piattaforma si perderebbe una potenziale parte di mercato e quindi di clienti. La seguente figura mostra, a scopi illustrativi, la frammentazione del mercato italiano nel 2016.

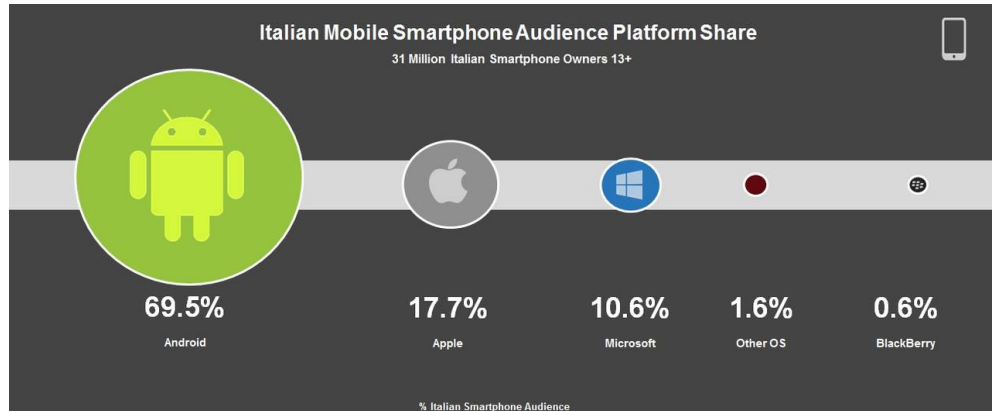


Figura 3.1: Frammentazione SO del mercato italiano nel 2016

Compresa la necessità di sviluppare più versioni della medesima applicazione in diverse piattaforme, il problema si sposta sulle risorse economiche e sul tempo che si ha a disposizione per lo sviluppo. Infatti lo sviluppo in differenti piattaforme comporta l'utilizzo di differenti linguaggi di programmazione e quindi la necessità di più programmatori esperti, precisamente almeno uno per piattaforma. Altre variabili di cui tener conto sono gli strumenti di sviluppo necessari, le API che si hanno a disposizione e fattori quali i sensori disponibili sui dispositivi, la dimensione degli schermi e le capacità di calcolo differenti.

L'obiettivo che i framework cross-platform cercano di raggiungere è la risoluzione di tutti questi problemi in maniera efficiente ed efficace in termini di risorse utilizzate, quindi, più precisamente, di ridurre gli effetti negativi della frammentazione del mercato.

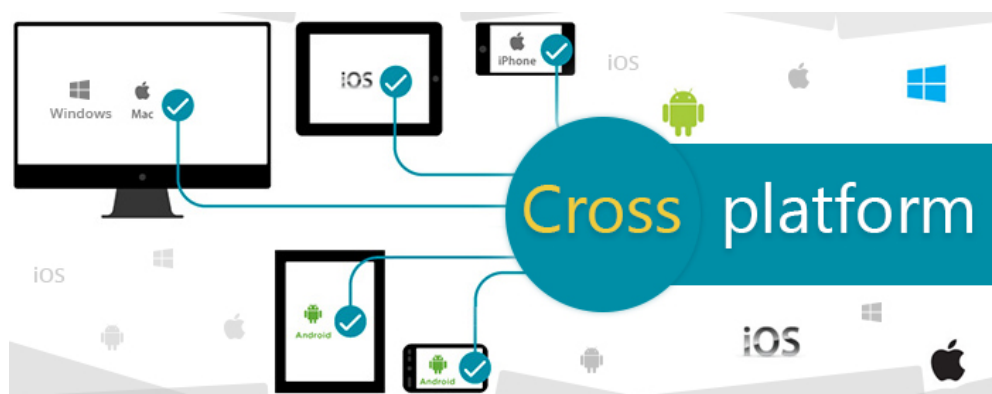


Figura 3.2: L'obiettivo dei framework cross-platform

Per raggiungere questo obiettivo i framework cross-platform permettono l'utilizzo di un solo linguaggio di programmazione, o di un insieme ristretto di linguaggi, per lo sviluppo di un unico codice sorgente che viene poi in secondo luogo convertito nel codice nativo delle piattaforme sulle quali si desidera distribuire l'applicazione.

Per concludere, dati oggettivi dimostrano che durante il 2016 l'utilizzo dei framework cross-platform ha portato ad un risparmio in termini di risorse economiche nell'80% dei casi e ad un risparmio di tempo nell'83% dei casi.

3.1.2 Approcci alla base dei framework cross-platform

Per la scelta del framework cross-platform più idoneo al problema è stato richiesto di studiare gli approcci secondo i quali i framework permettono la distribuzione su varie piattaforme. Esistono principalmente quattro approcci in base ai quali i framework possono essere classificati:

- * approccio web;
- * approccio ibrido;
- * approccio interpretato;
- * approccio cross-compiled.

In questa tesi vengono presentati solamente l'approccio ibrido e quello interpretato, poiché utilizzati dai framework proposti dal tutor aziendale.

L'approccio ibrido si interpone tra la realizzazione di un'applicazione web e lo sviluppo di un'applicazione mobile in codice nativo. In questo tipo di approccio l'applicazione viene sviluppata utilizzando tecnologie web ed eseguita all'interno di un container nativo sul dispositivo mobile. Per eseguire l'applicazione viene utilizzato il motore di rendering del browser del dispositivo mobile che si occupa di interpretare e visualizzare il contenuto HTML dell'applicazione tramite una visualizzazione web a schermo intero. L'accesso alle funzionalità native offerte dal dispositivo mobile è permesso grazie ad un livello astratto che si interpone tra l'applicazione ibrida e tali funzionalità. Questo livello astratto espone le funzionalità tramite API Javascript.

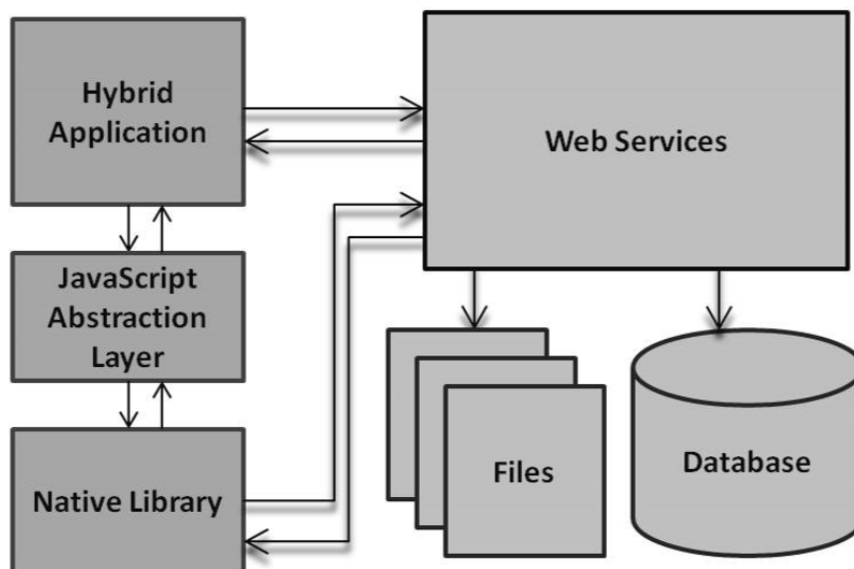


Figura 3.3: Architettura di un'applicazione ibrida

Nel caso delle applicazioni interpretate il codice sorgente dell'applicazione viene distribuito sul dispositivo mobile e in seguito interpretato da un interprete che si occupa di eseguire il codice a run-time. Anche in questo caso le funzionalità native vengono rese disponibili da un livello astratto. La caratteristica principale dell'interprete è che eseguendo il codice sorgente su differenti piattaforme, esso supporta lo sviluppo cross-platform delle applicazioni. L'applicazione interpretata interagisce con il livello astratto per accedere alle API native. Uno dei vantaggi dell'approccio interpretato è che utilizza elementi delle specifiche interfacce utente native per l'interazione utente. Infine, la logica applicativa è catturata in maniera del tutto indipendente dalla piattaforma sulla quale l'applicazione viene eseguita.

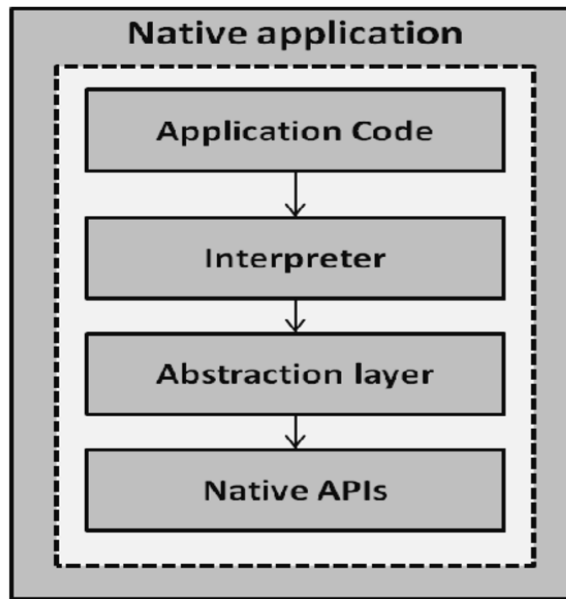


Figura 3.4: Architettura di un'applicazione interpretata

3.1.3 Xamarin

Si tratta di un framework cross-platform di proprietà dell'azienda Microsoft che utilizza due approcci differenti, l'approccio interpretato per l'ambiente Android e Windows, e l'approccio compilato per l'ambiente iOS. Più precisamente, per le piattaforme Android e Windows è possibile generare l'applicazione direttamente tramite i tool messi a disposizione dal framework e successivamente distribuirla sui rispettivi store, mentre per la piattaforma iOS è necessario un passo aggiuntivo. È richiesto, infatti, il passaggio per una macchina Apple che abbia installato XCode per eseguire la compilazione dell'applicazione. Infine, Xamarin richiede che venga utilizzato il linguaggio C# per lo sviluppo dell'applicazione. Nella figura sottostante viene presentata l'architettura di Xamarin.

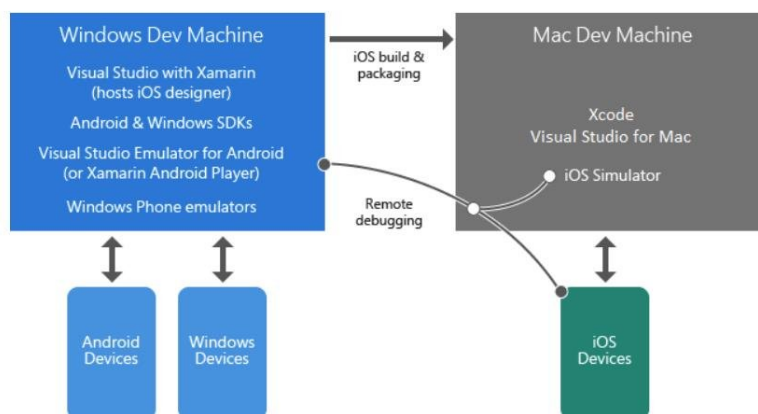


Figura 3.5: Architettura del framework Xamarin

3.1.4 PhoneGap

Si tratta di un framework cross-platform di proprietà dell'azienda Apache che utilizza un approccio di tipo ibrido. Quindi permette la realizzazione di applicazioni mobile tramite l'utilizzo di tecnologie web, che sono al giorno d'oggi strumenti conosciuti da tutti gli sviluppatori. L'accesso ai componenti hardware dei dispositivi mobile è permesso grazie all'utilizzo di plug-in scaricabili dalla pagina ufficiale del framework. Vantaggi importanti del framework sono la presenza di documentazione completa i plugin e la presenza di una community grande e sempre disponibile. Infine PhoneGap rende disponibili degli strumenti che facilitano lo sviluppo dell'applicazione: PhoneGap Desktop App, PhoneGap CLI, PhoneGap App e PhoneGap Build. I primi tre strumenti fanno parte dell'ambiente di sviluppo utilizzato durante lo stage e verranno descritti successivamente. PhoneGap Build è uno strumento che permette la build dell'applicazione direttamente su un server cloud Adobe a partire da un file zip contenente la cartella con il codice sorgente dell'applicazione. In seguito alla build è possibile generare e scaricare automaticamente l'applicazione per Windows o per Android. Per quanto riguarda iOS è necessario fornire i certificati richiesti da Apple per la distribuzione dell'applicazione. Nelle figure nella pagina successiva vengono presentate l'architettura di PhoneGap e una figura illustrativa di PhoneGap Build.

PhoneGap Architecture

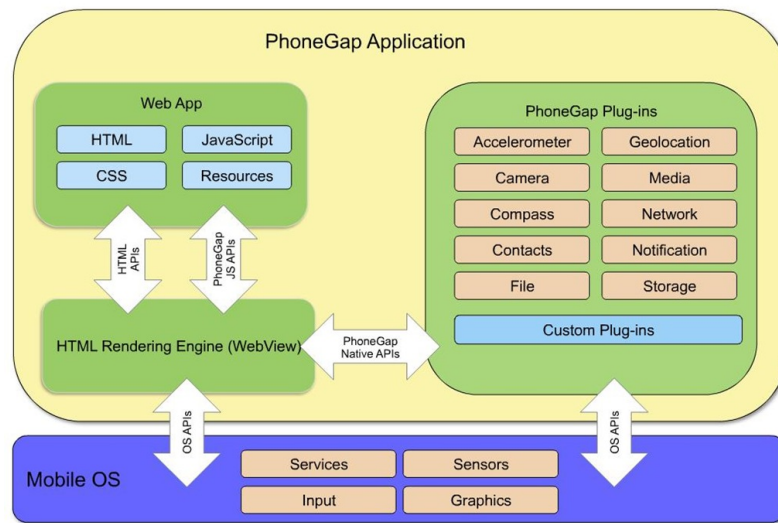


Figura 3.6: Architettura del framework PhoneGap

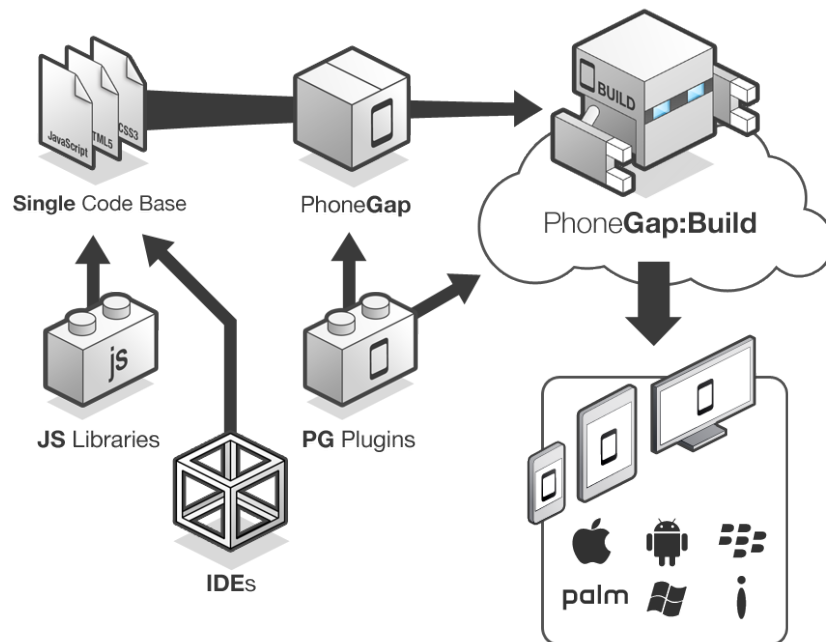


Figura 3.7: Figura illustrativa di PhoneGap Build

3.1.5 La scelta di PhoneGap

Lo stagista ha optato per il framework PhoneGap per tre motivi sostanziali:

1. **linguaggio di programmazione:** PhoneGap richiedeva l'utilizzo di tecnologie web, già conosciute e apprese dallo stagista all'Università. Lo studio del linguaggio C#, utilizzato da Xamarin, avrebbe richiesto un periodo di formazione che avrebbe sfiorato le 40 ore messe a disposizione per la formazione sulle tecnologie;
2. **linguaggio Javascript:** come detto precedentemente lo stagista era interessato ad approfondire il linguaggio Javascript, richiesto al giorno d'oggi dalla maggior parte delle aziende che si occupano della realizzazione di applicazioni web;
3. **facilità nello sviluppo dell'interfaccia grafica:** utilizzando tecnologie web risultava semplice progettare e sviluppare un'interfaccia grafica *responsive*_G, e quindi in grado di adattarsi alla maggior parte dei dispositivi mobile presenti sul mercato.

3.2 Ambiente di sviluppo

Durante il periodo di stage è stato utilizzato uno specifico ambiente di lavoro, comprendente tecnologie in parte imposte dal tutor aziendale e in parte scelte dallo stagista. La qualità di queste tecnologie ha impatto diretto sulla qualità di processo e quindi sulla qualità di prodotto, per cui è importante tenere l'ambiente di lavoro costantemente completo, ordinato e aggiornato. Per ottenere questo, lo stagista ha dovuto analizzare le tecnologie scelte per assicurarsi che fossero le più adatte per il dominio del problema.

3.2.1 Suite di PhoneGap

La suite di PhoneGap ha costituito parte fondamentale dell'ambiente di lavoro. Sono stati utilizzati i seguenti software della suite:

- * PhoneGap Desktop App: utilizzata inizialmente per prendere dimestichezza con il framework;
- * PhoneGap CLI: utilizzata dopo aver preso dimestichezza con il framework;
- * PhoneGap App: utilizzata inizialmente per testare l'applicazione generata dal framework.

PhoneGap Desktop App è un'applicazione installabile su Windows o Mac con la quale è possibile iniziare ad utilizzare il framework con estrema facilità. Essa fornisce un'interfaccia grafica per la creazione, la gestione e il testing di progetti PhoneGap. Per testare l'applicazione è sufficiente essere in possesso di un dispositivo Android con installata l'applicazione PhoneGap App. L'applicazione in versione iOS è stata rimossa dall'App Store per non aver passato i controlli di qualità Apple.

PhoneGap CLI estende le funzionalità offerte da PhoneGap Desktop App tramite un'interfaccia a riga di comando. Tale applicazione era lo strumento principale per l'utilizzo di PhoneGap prima della creazione dell'app desktop. PhoneGap CLI può essere utilizzata singolarmente o insieme a PhoneGap Desktop App e/o PhoneGap Build.

PhoneGap App è un'applicazione mobile, installabile solamente su dispositivi Android, con la quale è possibile testare l'applicazione PhoneGap senza generare ed installare nessun file applicazione. Per testare l'applicazione è sufficiente avviare l'esecuzione del progetto su PhoneGap Desktop App e collegare il computer di sviluppo e il dispositivo su cui è installata PhoneGap App alla stessa rete Wi-Fi. Dopo aver

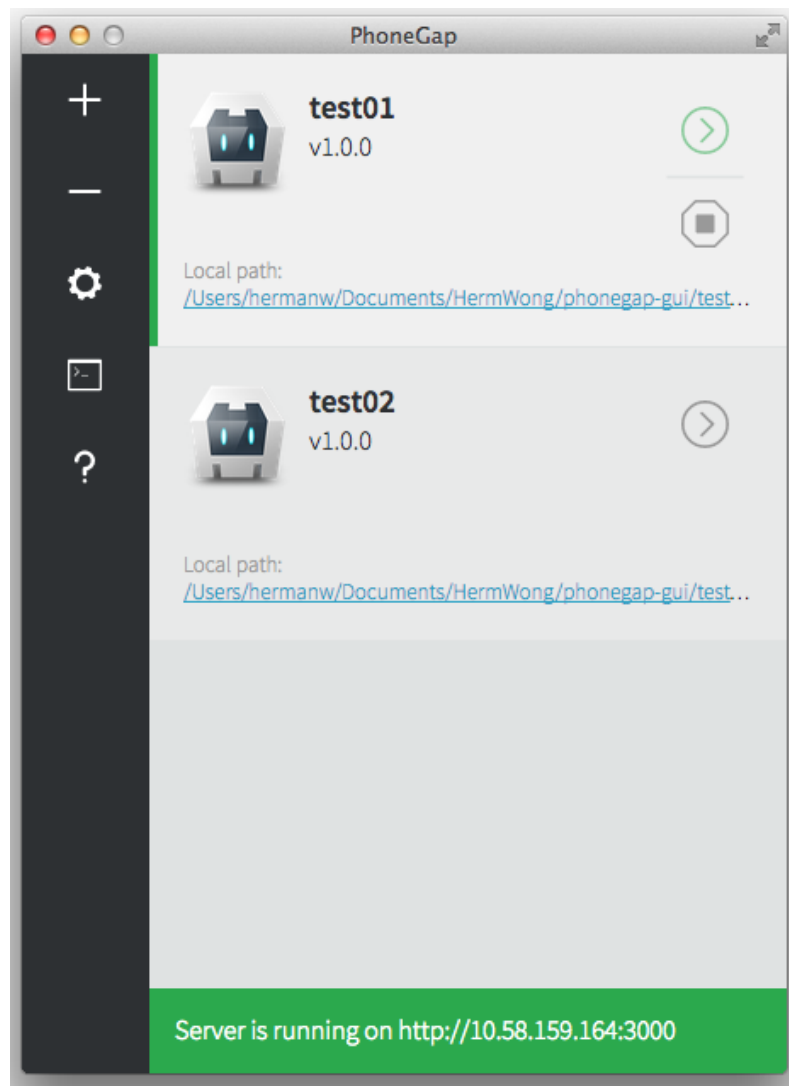


Figura 3.8: PhoneGap Desktop App

completato questi semplici passaggi viene richiesto di inserire su PhoneGap App l'indirizzo di rete su cui il progetto è stato messo in esecuzione da PhoneGap Desktop App. Dopo aver inserito questo indirizzo la reale applicazione verrà visualizzata sullo smartphone e sarà possibile testarla completamente.

3.2.2 Editor e IDE

3.2.2.1 Sublime Text 3.0

Per lo sviluppo del progetto PhoneGap è stato utilizzato l'editor Sublime Text 3.0, ritenuto dallo stagista il più semplice e leggero per la realizzazione di applicazioni web.



Figura 3.9: PhoneGap CLI e PhoneGap App



Figura 3.10: Logo di Sublime Text 3.0

3.2.2.2 Android Studio

Durante lo stage l'applicazione Android generata tramite i tool offerti dal framework PhoneGap non era soddisfacente, spesso non funzionava o l'interfaccia grafica non rispecchiava quella desiderata. Per cui è stato necessario installare Android Studio per applicare gli accorgimenti in codice nativo necessari a rendere l'app usabile. Android Studio è un ambiente di sviluppo integrato (IDE) basato sul software di JetBrains IntelliJ IDEA e progettato specificamente per lo sviluppo di applicazioni Android.



Figura 3.11: Logo di Android Studio

3.2.2.3 XCode

Durante lo stage l'applicazione iOS generata tramite i tool offerti dal framework PhoneGap non era soddisfacente, spesso non funzionava o l'interfaccia grafica non rispecchiava quella desiderata. Per cui è stato necessario installare XCode per applicare gli accorgimenti in codice nativo necessari a rendere l'app usabile. Xcode è un ambiente di sviluppo integrato (IDE) contenente una suite di strumenti utili allo sviluppo di software per i sistemi macOS, iOS, watchOS e tvOS. È completamente sviluppato e mantenuto da Apple.

3.2.2.4 Eclipse JEE

Durante lo stage è stata richiesta la realizzazione di un servizio web che si interponesse tra la logica applicativa di moviORDER e il database presente sul server Azure di VisioneImpresa. Il servizio web è stato realizzato in linguaggio Java tramite l'utilizzo di oggetti servlet. Per la realizzazione degli oggetti servlet è stato utilizzato l'IDE Eclipse JEE che offre buoni strumenti per lo sviluppo di applicazioni web Java. Eclipse è un ambiente di sviluppo integrato multi-linguaggio e multipiattaforma, ideato da un consorzio di grandi società quali Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP e Serena Software, chiamato Eclipse Foundation.



Figura 3.12: Logo di XCode



Figura 3.13: Logo di Eclipse

3.2.3 Gestione DBMS

Durante il progetto è stato utilizzato il DBMS Microsoft SQL Server. Per una gestione veloce ed ottimale dello stesso si è deciso di usare il software SQL Server Management Studio. SQL Server Management Studio (SSMS) è un'applicazione software usata per configurare, gestire e amministrare database, in locale o su un server cloud, con il DBMS Microsoft SQL Server.

3.2.4 Server web

Per lo sviluppo del progetto è stato realizzato un servizio web che fornisce una API all'applicazione per poter accedere ai dati contenuti nel database sul server cloud di VisioneImpresa. Per permettere l'esecuzione del servizio web sul server cloud si è utilizzato Apache Tomcat. Apache Tomcat (o semplicemente Tomcat) è un web server

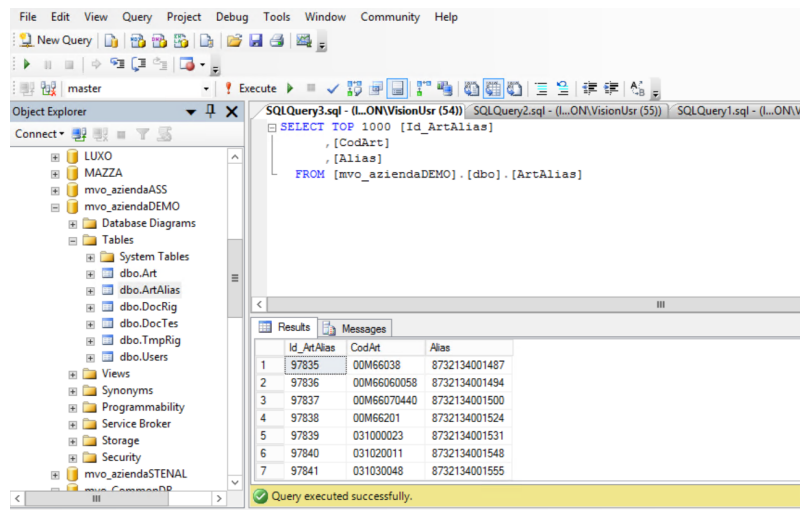


Figura 3.14: Screenshot di SQL Server Management Studio

(nella forma di contenitore servlet) open source sviluppato dalla Apache Software Foundation. Implementa le specifiche JavaServer Pages (JSP) e servlet, fornendo quindi una piattaforma software per l'esecuzione di applicazioni Web sviluppate in linguaggio Java.

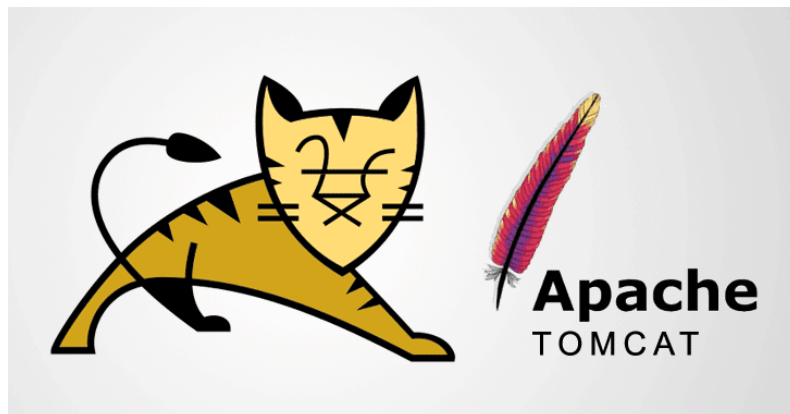


Figura 3.15: Logo di Apache Tomcat

3.2.5 Cloud computing

Il database con il quale l'applicazione interagisce per offrire le funzionalità all'utente risiede su un server cloud di VisioneImpresa. Tale server è di proprietà di Microsoft Azure. Microsoft Azure (precedentemente nota come Windows Azure) è la piattaforma cloud pubblica di Microsoft, che offre servizi di cloud computing. Tramite Azure vengono erogati servizi appartenenti a diverse categorie quali: risorse di elaborazione, archiviazione e memorizzazione dati, trasmissione dati e interconnessione di reti,

analisi, intelligence, apprendimento automatico, sicurezza e gestione delle identità, monitoraggio e gestione, nonché servizi per lo sviluppo di applicazioni.



Figura 3.16: Logo di Microsoft Azure

3.2.6 Strumenti di testing

Durante lo sviluppo di *moviORDER* sono stati eseguiti dei test per verificare il corretto funzionamento del servizio web e dell'applicazione. Per il test delle API del servizio web si è utilizzato Postman. Postman è uno strumento di API testing che permette di testare API direttamente, o come parte di test d'integrazione, per determinare se soddisfano criteri di funzionalità, affidabilità, performance e sicurezza. Nel caso di *moviORDER*, il test avviene tramite l'invio di una richiesta HTTP POST, con parametri impostabili, alla API sul server cloud. In seguito alla richiesta il software visualizza il JSON restituito dalla API e lo sviluppatore può verificare se la risposta del servizio è quella attesa.

Per testare il corretto funzionamento dell'applicazione si è utilizzata la console del browser Google Chrome facente parte degli strumenti offerti dallo stesso per gli sviluppatori web. Tramite la console è stato possibile verificare la logica applicativa di *moviORDER*, poiché scritta in JavaScript.

3.2.7 Strumenti di versioning e ticketing

Per scelta dello stagista il progetto è stato sottoposto a controllo di versione in ogni sua parte: applicazione, servizio web e documentazione. Questo ha permesso, principalmente nel caso dell'applicazione, di tornare a *baseline_G* sicure nel caso di sovrascritture o perdite accidentali, o commit di codice con errori di programmazione. Per il controllo di versione si è utilizzato lo strumento Git e in particolare il servizio di hosting GitHub. Lo stagista ha scelto tali software perché già utilizzati in progetti didattici durante l'Università.

Lo stagista ha scelto inoltre di utilizzare uno strumento di ticketing per facilitare la pianificazione del progetto. Lo strumento scelto è Asana. Asana è un'applicazione web



Figura 3.17: Logo di Postman

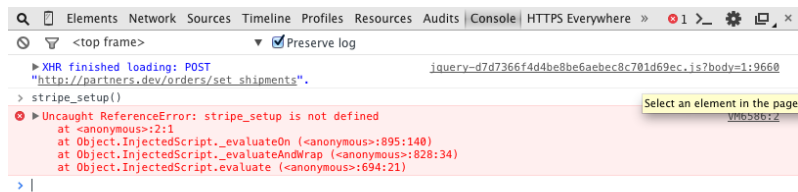


Figura 3.18: Screenshot della console di Google Chrome

e mobile progettata per aiutare i team ad organizzare, tracciare e gestire il loro lavoro. In particolare lo strumento è stato utilizzato per dare una scadenza ai ticket in base alla pianificazione decisa insieme al tutor aziendale all'inizio del periodo di stage.

3.2.8 Strumenti di modellazione e documentazione

Il progetto ha richiesto lo sviluppo di diagrammi di Gantt in fase di pianificazione e di diagrammi UML in fase di analisi dei requisiti e di progettazione. Per la costruzione dei diagrammi di Gantt è stato utilizzato il software Gantt Project, mentre per la costruzione dei diagrammi UML dei casi d'uso e delle classi è stato utilizzato il software Astah UML. I software sono stati scelti perché già appresi durante il progetto di Ingegneria del Software.

Per la stesura della documentazione si è utilizzato l'editor TexMaker, anch'esso utilizzato durante il progetto di Ingegneria del Software. TexMaker permette l'inserimento veloce dei comandi \LaTeX più utilizzati tramite la pressione di pulsanti sull'interfaccia

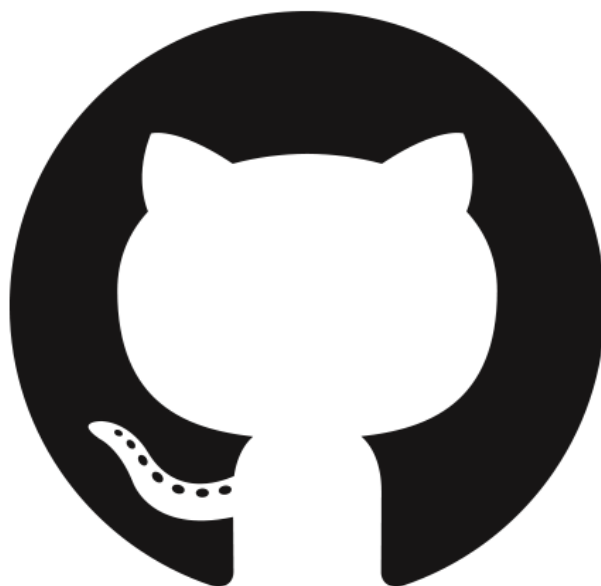


Figura 3.19: Logo di GitHub



Figura 3.20: Logo di Asana

grafica, l'integrazione con un dizionario per il controllo ortografico e la compilazione e visione del pdf prodotto.



Figura 3.21: Logo di Gantt Project e Astah



Figura 3.22: Logo di TexMaker

3.2.9 Linguaggi di programmazione e markup

3.2.9.1 HTML5, CSS3 e JavaScript

3.2.9.2 Java

3.2.9.3 \LaTeX

3.2.10 DBMS

scrivere di sql server

Capitolo 4

Analisi dei requisiti

Questo capitolo descrive i casi d'uso e i requisiti della piattaforma moviORDER, individuati e classificati per definire nel dettaglio obiettivi e funzionalità del sistema. I casi d'uso e i requisiti sono stati dedotti da un'analisi preliminare eseguita dal tutor aziendale, la quale è stata perfezionata dallo stagista per perseguire massima efficienza ed efficacia del sistema. Le convenzioni adottate per il codice univoco di casi d'uso e requisiti sono presentate in [Appendice §A](#).

4.1 Casi d'uso

La descrizione di ogni caso d'uso si compone dei seguenti punti:

1. **attori:** lista di attori principali e secondari coinvolti nel caso d'uso;
2. **descrizione:** breve descrizione del caso d'uso;
3. **pre-condizioni:** condizioni che devono essere verificate prima dell'esecuzione del caso d'uso;
4. **post-condizioni:** condizioni che devono essere verificate dopo dell'esecuzione del caso d'uso;
5. **scenario principale:** descrizione composta dal flusso dei sottocasi d'uso del caso d'uso principale;
6. **scenari alternativi:** descrizione composta dai casi d'uso che non appartengono al flusso principale di esecuzione del caso d'uso (estensioni), se presenti;
7. **specializzazioni:** indica quali sono tutte le specializzazioni del caso d'uso, se presenti;
8. **generalizzazioni:** indica quali sono tutte le generalizzazioni del caso d'uso, se presenti.

Per lo studio dei casi di utilizzo più complessi della piattaforma, sono stati creati dei diagrammi dei casi d'uso che meglio descrivono funzioni e/o servizi offerti dal sistema, così come sono percepiti e utilizzati dagli attori che interagiscono con lo stesso. Per la definizione dei diagrammi UML dei casi d'uso è stato utilizzato lo standard *UML 2.0G*.

4.1.1 Attori del sistema

Lo scopo di moviORDER è permettere alle aziende che forniscono prodotti di vendere gli stessi ai propri clienti tramite un'applicazione multiplatforma. MoviORDER viene distribuita da VisioneImpresa alle aziende che forniscono prodotti, la quale viene distribuita dalle aziende stesse ai propri clienti. Gli utilizzatori finali di moviORDER sono quindi i clienti delle singole aziende clienti di VisioneImpresa. L'accesso all'applicazione è consentito solamente agli utenti provvisti di credenziali di accesso, le quali vengono distribuite, insieme all'applicazione, dal fornitore. Non è prevista quindi una funzionalità di registrazione. Nel contesto di moviORDER vi sono quindi due tipologie di attori:

1. **Utente non autenticato:** è un utente che non ha effettuato l'accesso al sistema, al quale viene offerta la sola funzionalità di autenticazione. Una volta che un utente non autenticato viene riconosciuto accedendo al sistema, diventa un utente autenticato;
2. **Utente autenticato:** è un utente che ha effettuato l'accesso al sistema e che può usufruire di tutte le sue funzionalità. Le funzionalità offerte all'utente autenticato sono:
 - * possibilità di effettuare il logout;
 - * possibilità di aggiungere articoli al proprio carrello;
 - * possibilità di modificare gli articoli nel proprio carrello;
 - * possibilità di rimuovere articoli dal proprio carrello;
 - * possibilità di inviare un ordine alla propria azienda.

4.1.2 UC1 - Azioni utente non autenticato

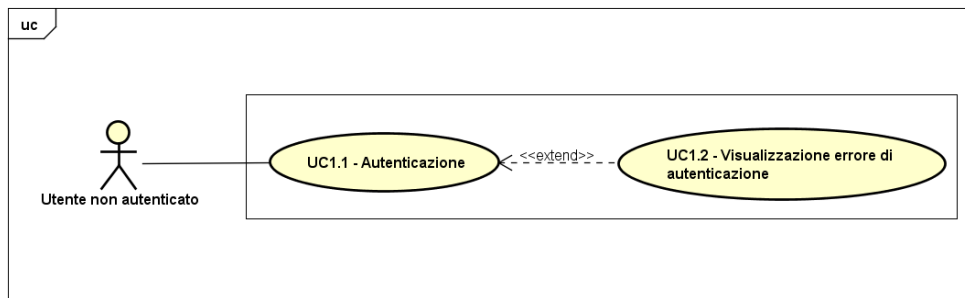


Figura 4.1: Use Case - UC1: Azioni utente non autenticato

- * **Attore:** Utente non autenticato;
- * **Descrizione:** L'attore può eseguire l'operazione di autenticazione alla piattaforma moviORDER;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione e non è ancora stato riconosciuto dal sistema;

- * **Post-condizioni:** L'attore ha eseguito l'operazione che desiderava compiere da utente non autenticato;
- * **Scenario principale:** UC1.1 - Autenticazione;
- * **Scenario alternativo:** L'attore ha fornito credenziali di accesso non corrispondenti a nessun utente registrato dall'azienda, oppure non riesce ad accedere al sistema perché è stato bloccato dall'azienda stessa: UC1.2 - Visualizzazione errore di autenticazione.

4.1.3 UC1.1 - Autenticazione

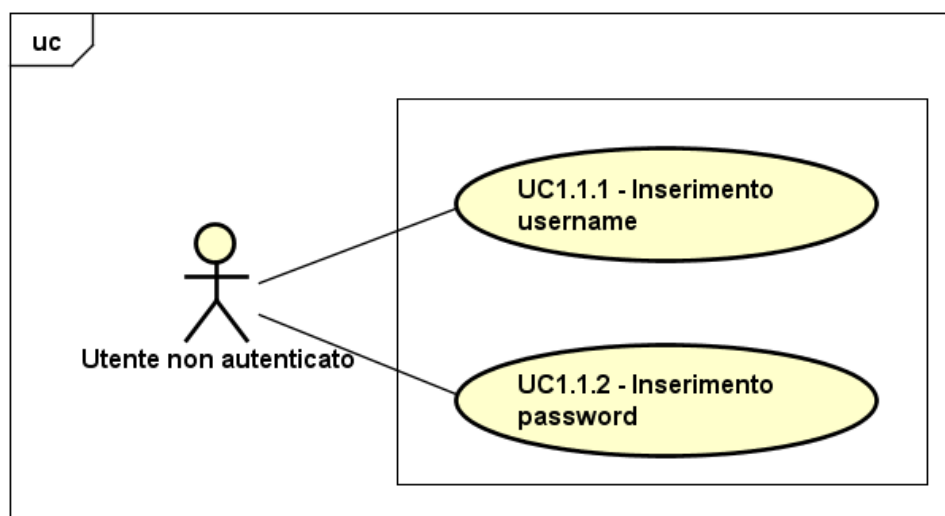


Figura 4.2: Use Case - UC1.1: Autenticazione

- * **Attore:** Utente non autenticato;
- * **Descrizione:** L'attore può eseguire l'operazione di autenticazione;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, non è ancora riconosciuto dal sistema e ha espresso la volontà di effettuare l'autenticazione a moviORDER;
- * **Post-condizioni:** L'attore ha eseguito l'operazione di accesso al sistema ed è quindi ora riconosciuto come utente autenticato;
- * **Scenario principale:**
 1. UC1.1.1 - Inserimento username;
 2. UC1.1.2 - Inserimento password.

4.1.4 UC1.1.1 - Inserimento username

- * **Attore:** Utente non autenticato;
- * **Descrizione:** L'attore deve inserire una username per l'operazione di autenticazione;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, non è ancora riconosciuto dal sistema e il sistema richiede l'inserimento di una username per l'operazione di autenticazione;
- * **Post-condizioni:** L'attore ha inserito la username per l'operazione di autenticazione;
- * **Scenario principale:** L'attore inserisce la username tramite una text box.

4.1.5 UC1.1.2 - Inserimento password

- * **Attore:** Utente non autenticato;
- * **Descrizione:** L'attore deve inserire una password per l'operazione di autenticazione;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, non è ancora riconosciuto dal sistema e il sistema richiede l'inserimento di una password per l'operazione di autenticazione;
- * **Post-condizioni:** L'attore ha inserito la password per l'operazione di autenticazione;
- * **Scenario principale:** L'attore inserisce la password tramite una text box.

4.1.6 UC1.2 - Visualizzazione errore di autenticazione

- * **Attore:** Utente non autenticato;
- * **Descrizione:** L'attore inserisce credenziali di accesso non corrispondenti a nessun utente registrato dall'azienda, oppure l'utente è stato bloccato dall'azienda stessa;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, non è ancora riconosciuto dal sistema e inserisce credenziali di accesso non corrispondenti a nessun utente registrato dall'azienda, oppure è stato bloccato;
- * **Post-condizioni:** L'attore ha visualizzato un messaggio d'errore relativo all'impossibilità di eseguire l'autenticazione per inserimento di credenziali errate o perché è stato bloccato;
- * **Scenario principale:** L'attore visualizza un messaggio d'errore relativo all'impossibilità di eseguire l'autenticazione per inserimento di credenziali errate o perché è stato bloccato.

4.1.7 UC2 - Azioni utente autenticato

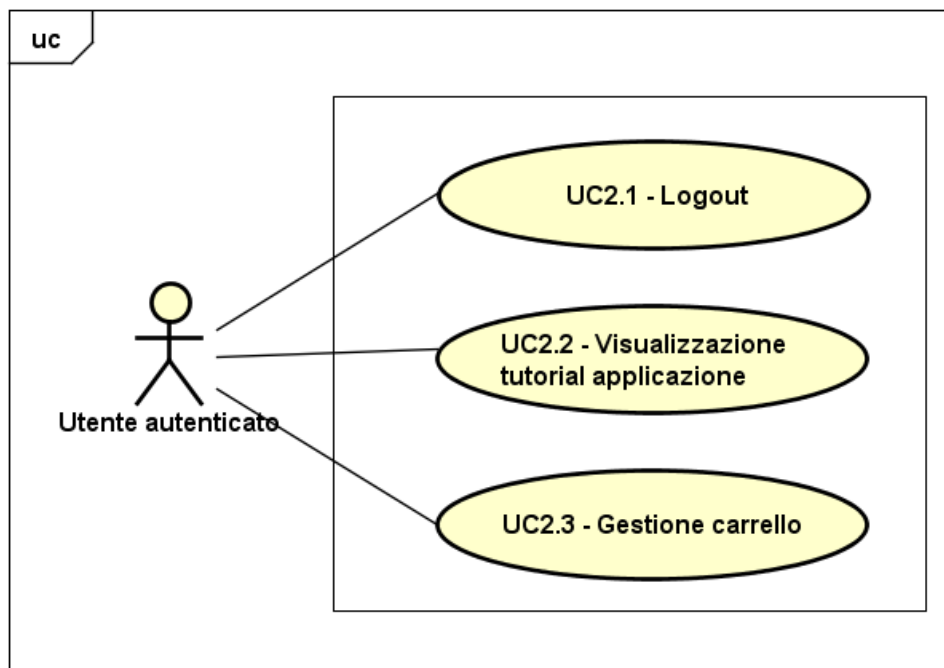


Figura 4.3: Use Case - UC2: Azioni utente autenticato

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può:
 1. eseguire l'operazione di logout;
 2. visualizzare il tutorial dell'applicazione;
 3. gestire il proprio carrello.
- * **Pre-condizioni:** L'attore ha avviato l'applicazione ed è riconosciuto dal sistema;
- * **Post-condizioni:** L'attore ha eseguito le azioni che desiderava compiere da utente autenticato;
- * **Scenario principale:**
 1. UC2.1 - Logout;
 2. UC2.2 - Visualizzazione tutorial applicazione;
 3. UC2.3 - Gestione carrello.

4.1.8 UC2.1 - Logout

- * **Attore:** Utente autenticato;

- * **Descrizione:** L'attore può eseguire l'operazione di logout;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha espresso la volontà di effettuare il logout da moviORDER;
- * **Post-condizioni:** L'attore ha eseguito l'operazione di logout da moviORDER ed è quindi uscito dal sistema tornando ad essere un utente non autenticato;
- * **Scenario principale:** L'attore esegue l'operazione di logout da moviORDER premendo sul relativo bottone.

4.1.9 UC2.2 - Visualizzazione tutorial applicazione

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può visualizzare il tutorial dell'applicazione;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto la visualizzazione del tutorial dell'applicazione;
- * **Post-condizioni:** L'attore ha visualizzato il tutorial dell'applicazione;
- * **Scenario principale:** L'attore visualizza il tutorial dell'applicazione premendo sul relativo bottone.

4.1.10 UC2.3 - Gestione carrello

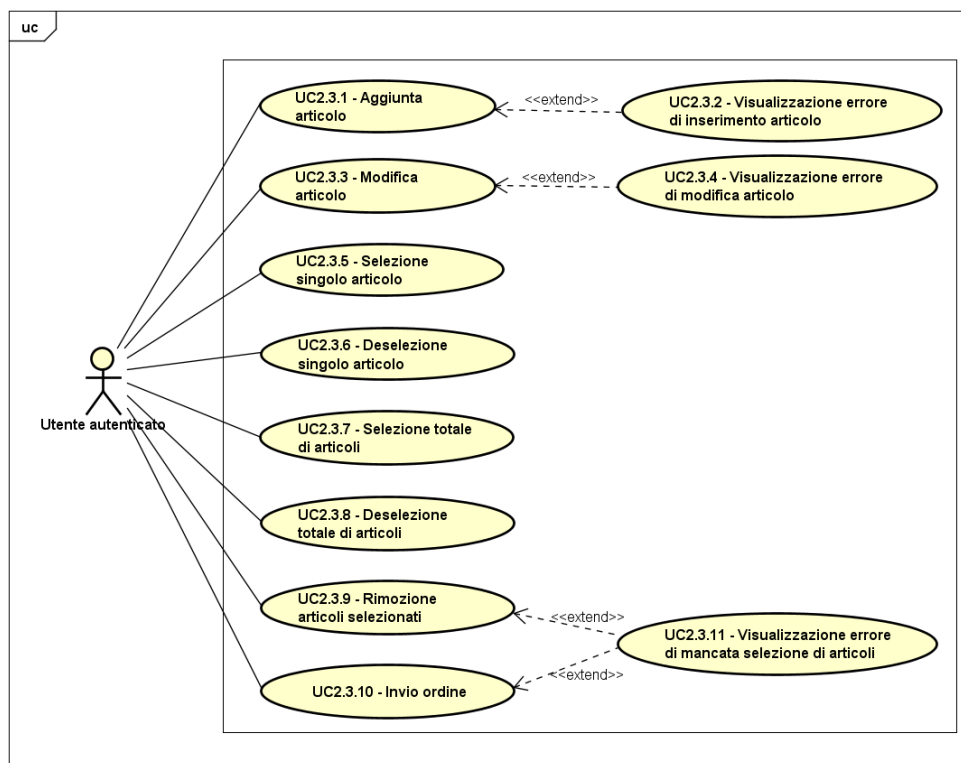


Figura 4.4: Use Case - UC2.3: Gestione carrello

* **Attore:** Utente autenticato;

* **Descrizione:** L'attore può:

1. aggiungere un articolo al carrello;
2. modificare un articolo in carrello;
3. selezionare un singolo articolo in carrello;
4. deselectare un singolo articolo in carrello;
5. selezionare tutti gli articoli in carrello;
6. deselectare tutti gli articoli in carrello;
7. rimuovere articoli dal carrello;
8. inviare un ordine.

* **Pre-condizioni:** L'attore ha avviato l'applicazione ed è riconosciuto dal sistema;

* **Post-condizioni:** L'attore ha eseguito le operazioni che desiderava compiere sul proprio carrello;

* **Scenario principale:**

1. UC2.3.1 - Aggiunta articolo;
2. UC2.3.3 - Modifica articolo;
3. UC2.3.5 - Selezione singolo articolo;
4. UC2.3.6 - Deselezione singolo articolo;
5. UC2.3.7 - Selezione totale di articoli;
6. UC2.3.8 - Deselezione totale di articoli;
7. UC2.3.9 - Rimozione articoli selezionati;
8. UC2.3.10 - Invio ordine.

*** Scenari alternativi:**

- Durante l'operazione di aggiunta articolo, l'attore ha inserito un codice articolo non corrispondente a nessun articolo venduto dall'azienda, oppure ha inserito una quantità non permessa per l'articolo, oppure la scansione del codice a barre per la ricerca del codice articolo è fallita: UC2.3.2 - Visualizzazione errore di inserimento articolo;
- Durante l'operazione di modifica articolo, l'attore ha inserito una quantità non permessa per l'articolo: UC2.3.4 - Visualizzazione errore di modifica articolo;
- L'attore ha premuto il bottone di rimozione articoli selezionati senza aver selezionato alcun articolo dal carrello: UC2.3.11 - Visualizzazione errore di mancata selezione di articoli;
- L'attore ha premuto il bottone di invio ordine senza aver selezionato alcun articolo dal carrello: UC2.3.12 - Visualizzazione errore di mancata selezione di articoli.

4.1.11 UC2.3.1 - Aggiunta articolo

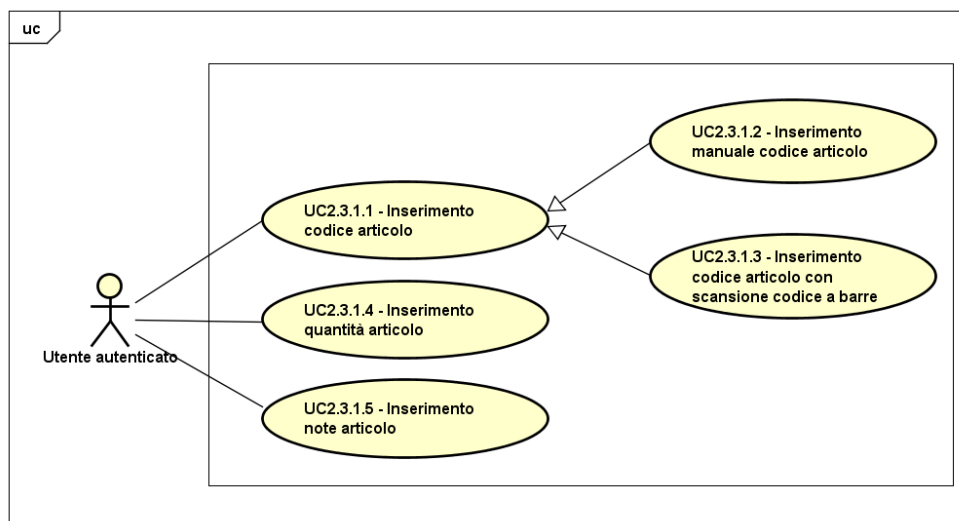


Figura 4.5: Use Case - UC2.3.1: Aggiunta articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può aggiungere un articolo al carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di aggiungere un articolo al carrello;
- * **Post-condizioni:** L'attore ha aggiunto l'articolo al carrello;
- * **Scenario principale:**
 1. UC2.3.1.1 - Inserimento codice articolo;
 2. UC2.3.1.4 - Inserimento quantità articolo;
 3. UC2.3.1.5 - Inserimento note articolo.

4.1.12 UC2.3.1.1 - Inserimento codice articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore deve inserire un codice articolo per l'operazione di aggiunta articolo;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di aggiungere un articolo in carrello e il sistema richiede l'inserimento di un codice articolo per l'operazione di aggiunta articolo;
- * **Post-condizioni:** L'attore ha inserito il codice articolo;
- * **Scenario principale:** L'attore inserisce il codice articolo manualmente (UC2.3.1.2) o tramite la scansione di un codice a barre (UC2.3.1.3);

* **Specializzazioni:**

1. UC2.3.1.2 - Inserimento manuale codice articolo;
2. UC2.3.1.3 - Inserimento codice articolo con scansione codice a barre.

4.1.13 UC2.3.1.2 - Inserimento manuale codice articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore deve inserire un codice articolo per l'operazione di aggiunta articolo;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di aggiungere un articolo in carrello e quando il sistema richiede l'inserimento di un codice articolo per l'operazione di aggiunta articolo, l'attore esprime la volontà di voler inserire il codice articolo manualmente;
- * **Post-condizioni:** L'attore ha inserito il codice articolo manualmente;
- * **Scenario principale:** L'attore inserisce il codice articolo manualmente tramite una text box;
- * **Generalizzazione:** UC2.3.1.1 - Inserimento codice articolo.

4.1.14 UC2.3.1.3 - Inserimento codice articolo con scansione codice a barre

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore deve inserire un codice articolo per l'operazione di aggiunta articolo;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di aggiungere un articolo in carrello e quando il sistema richiede l'inserimento di un codice articolo per l'operazione di aggiunta articolo, l'attore esprime la volontà di voler inserire il codice articolo mediante la scansione del codice a barre di un articolo;
- * **Post-condizioni:** L'attore ha inserito il codice articolo mediante la scansione del codice a barre dell'articolo;
- * **Scenario principale:** L'attore inserisce il codice articolo mediante la scansione del codice a barre dell'articolo;
- * **Generalizzazione:** UC2.3.1.1 - Inserimento codice articolo.

4.1.15 UC2.3.1.4 - Inserimento quantità articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore deve inserire una quantità per l'operazione di aggiunta articolo;

- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di aggiungere un articolo in carrello e il sistema richiede l'inserimento di una quantità per l'operazione di aggiunta articolo;
- * **Post-condizioni:** L'attore ha inserito la quantità dell'articolo;
- * **Scenario principale:** L'attore inserisce la quantità dell'articolo tramite una text box.

4.1.16 UC2.3.1.5 - Inserimento note articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può inserire delle note durante l'operazione di aggiunta articolo;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di aggiungere un articolo in carrello e il sistema permette l'inserimento di note durante l'operazione di aggiunta articolo;
- * **Post-condizioni:** L'attore ha inserito delle note per l'articolo;
- * **Scenario principale:** L'attore inserisce delle note per l'articolo tramite una text area.

4.1.17 UC2.3.2 - Visualizzazione errore di inserimento articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore inserisce un codice articolo non corrispondente a nessun articolo venduto dall'azienda, oppure inserisce una quantità non permessa per l'articolo, oppure la scansione del codice a barre per la ricerca del codice articolo è fallita;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di aggiungere un articolo in carrello e ha inserito un codice articolo che non corrisponde a nessun articolo venduto dall'azienda, oppure ha inserito una quantità non permessa per l'articolo, oppure la scansione del codice a barre per la ricerca del codice articolo è fallita;
- * **Post-condizioni:** L'attore ha visualizzato un messaggio d'errore relativo all'impossibilità di aggiungere l'articolo in carrello;
- * **Scenario principale:** L'attore visualizza un messaggio d'errore relativo all'impossibilità di aggiungere l'articolo in carrello.

4.1.18 UC2.3.3 - Modifica articolo

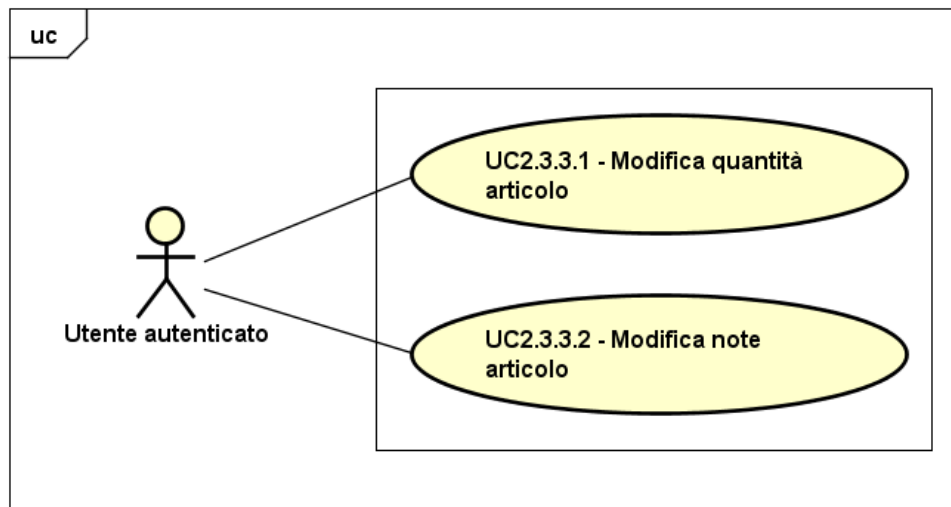


Figura 4.6: Use Case - UC2.3.3: Modifica articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può modificare un articolo in carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di modificare un articolo in carrello;
- * **Post-condizioni:** L'attore ha modificato l'articolo in carrello;
- * **Scenario principale:**
 1. UC2.3.3.1 - Modifica quantità articolo;
 2. UC2.3.3.2 - Modifica note articolo.

4.1.19 UC2.3.3.1 - Modifica quantità articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può modificare la quantità dell'articolo selezionato;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di modificare un articolo in carrello e il sistema permette la modifica della quantità dell'articolo selezionato;
- * **Post-condizioni:** L'attore ha modificato la quantità dell'articolo selezionato;
- * **Scenario principale:** L'attore modifica la quantità dell'articolo selezionato tramite una text box.

4.1.20 UC2.3.3.2 - Modifica note articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può modificare le note dell'articolo selezionato;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di modificare un articolo in carrello e il sistema permette la modifica della note dell'articolo selezionato;
- * **Post-condizioni:** L'attore ha modificato le note dell'articolo selezionato;
- * **Scenario principale:** L'attore modifica le note dell'articolo selezionato tramite una text box.

4.1.21 UC2.3.4 - Visualizzazione errore di modifica articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore inserisce una quantità non permessa per l'articolo selezionato;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di modificare un articolo in carrello e ha inserito una quantità non permessa per l'articolo selezionato;
- * **Post-condizioni:** L'attore ha visualizzato un messaggio d'errore relativo all'impossibilità di modificare l'articolo selezionato;
- * **Scenario principale:** L'attore visualizza un messaggio d'errore relativo all'impossibilità di modificare l'articolo selezionato.

4.1.22 UC2.3.5 - Selezione singolo articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può selezionare un articolo non ancora selezionato in carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di selezionare un articolo non ancora selezionato in carrello;
- * **Post-condizioni:** L'attore ha selezionato l'articolo in carrello;
- * **Scenario principale:** L'attore seleziona l'articolo in carrello tramite una checkbox.

4.1.23 UC2.3.6 - Deselezione singolo articolo

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può deselezionare un articolo selezionato in carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di deselezionare un articolo selezionato in carrello;

- * **Post-condizioni:** L'attore ha deselezionato l'articolo in carrello;
- * **Scenario principale:** L'attore deseleziona l'articolo in carrello tramite una check-box.

4.1.24 UC2.3.7 - Selezione totale di articoli

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può selezionare tutti gli articoli in carrello non ancora selezionati;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di selezionare tutti gli articoli in carrello non ancora selezionati;
- * **Post-condizioni:** L'attore ha selezionato tutti gli articoli in carrello non ancora selezionati;
- * **Scenario principale:** L'attore seleziona tutti gli articoli in carrello non ancora selezionati premendo su un bottone.

4.1.25 UC2.3.8 - Deselezione totale di articoli

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può deselezionare tutti gli articoli in carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di deselezionare tutti gli articoli in carrello;
- * **Post-condizioni:** L'attore ha deselezionato tutti gli articoli in carrello;
- * **Scenario principale:** L'attore deseleziona tutti gli articoli in carrello premendo su un bottone.

4.1.26 UC2.3.9 - Rimozione articoli selezionati

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può rimuovere gli articoli selezionati dal carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di rimuovere gli articoli selezionati dal carrello;
- * **Post-condizioni:** L'attore ha rimosso gli articoli selezionati dal carrello;
- * **Scenario principale:** L'attore rimuove gli articoli selezionati dal carrello premendo su un bottone.

4.1.27 UC2.3.10 - Invio ordine

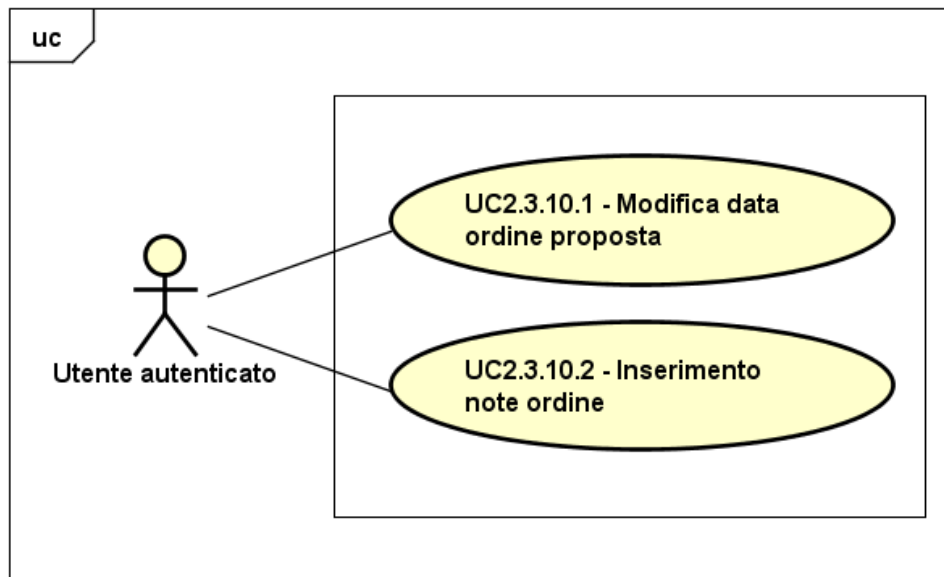


Figura 4.7: Use Case - UC2.3.10: Invio ordine

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può inviare un ordine composto dagli articoli selezionati in carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema e ha richiesto di inviare un ordine composto dagli articoli selezionati in carrello;
- * **Post-condizioni:** L'attore ha inviato un ordine composto dagli articoli selezionati in carrello;
- * **Scenario principale:**
 1. UC2.3.10.1 - Modifica data ordine proposta;
 2. UC2.3.10.2 - Inserimento note ordine.

4.1.28 UC2.3.10.1 - Modifica data ordine proposta

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può modificare la data d'ordine proposta;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di inviare un ordine e il sistema permette la modifica della data d'ordine proposta;
- * **Post-condizioni:** L'attore ha modificato la data d'ordine proposta;
- * **Scenario principale:** L'attore modifica la data d'ordine proposta tramite una text box.

4.1.29 UC2.3.10.2 - Inserimento note ordine

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore può delle note per l'ordine;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di inviare un ordine e il sistema permette l'inserimento di note per l'ordine;
- * **Post-condizioni:** L'attore ha inserito le note per l'ordine;
- * **Scenario principale:** L'attore inserisce le note per l'ordine tramite una text area.

4.1.30 UC2.3.11 - Visualizzazione errore di mancata selezione di articoli

- * **Attore:** Utente autenticato;
- * **Descrizione:** L'attore preme sul bottone di rimozione articoli o invio ordine senza aver selezionato degli articoli in carrello;
- * **Pre-condizioni:** L'attore ha avviato l'applicazione, è riconosciuto dal sistema, ha richiesto di rimuovere degli articoli dal carrello o di inviare un ordine, senza aver effettivamente selezionato degli articoli in carrello;
- * **Post-condizioni:** L'attore ha visualizzato un messaggio d'errore relativo all'impossibilità di rimuovere gli articoli selezionati o inviare un ordine composto dagli articoli selezionati;
- * **Scenario principale:** L'attore visualizza un messaggio d'errore relativo all'impossibilità di rimuovere gli articoli selezionati o inviare un ordine composto dagli articoli selezionati.

4.2 Requisiti

In questa sezione sono presentati i requisiti della piattaforma moviORDER, dedotti dai diagrammi dei casi d'uso, dal capitolato fornito dal tutor aziendale e da raffinamenti di questo, ottenuti in seguito a brevi interazioni avvenute con il tutor aziendale. I requisiti sono stati classificati dallo stagista in base alla loro tipologia e importanza strategica. Sono stati individuati:

1. **requisiti funzionali:** descrivono funzionalità del sistema;
2. **requisiti qualitativi:** descrivono qualità del prodotto finale;
3. **requisiti di vincolo:** descrivono vincoli imposti dal tutor aziendale o dal dominio del problema.

L'importanza strategica dei requisiti individuati è stata negoziata con il tutor aziendale e si è deciso che ciascun requisito deve appartenere ad una delle seguenti categorie:

- * **obbligatorio:** non più negoziabile e irrinunciabile per il tutor aziendale;
- * **desiderabile:** rappresenta un valore aggiunto d'interesse riconosciuto da parte del tutor aziendale;
- * **facoltativo:** relativamente utile per gli scopi dell'applicazione.

Nelle seguenti sezioni vengono dettagliati i requisiti di moviORDER. In ogni sezione è presente una tabella contenente per ogni requisito:

- * **codice identificativo:** codice che identifica univocamente il requisito, utile a fini di tracciabilità;
- * **descrizione:** breve descrizione testuale del requisito;
- * **fonte:** fonte da cui è stato dedotto il requisito, utile a fini di tracciabilità. La fonte di un requisito può essere:
 - **caso d'uso:** se il requisito è stato dedotto da un caso d'uso specifico;
 - **capitolato:** se il requisito è stato dedotto dal capitolato d'appalto;
 - **stagista:** se il requisito è stato introdotto dallo stagista durante il periodo di stage;
 - **tutor:** se il requisito è stato introdotto dal tutor aziendale durante il periodo di stage.

4.2.1 Requisiti funzionali

Id Requisito	Descrizione	Fonti
RFO1	Se l'applicazione viene lanciata in assenza di connettività, deve essere visualizzato un messaggio d'errore di connettività assente e l'applicazione deve essere chiusa	Stagista
RFO2	L'utente non autenticato deve poter accedere all'applicazione	Capitolato UC1.1
RFO2.1	Per accedere, l'utente non autenticato deve poter inserire una username	Capitolato UC1.1.1
RFO2.2	Per accedere, l'utente non autenticato deve poter inserire una password	Capitolato UC1.1.2
RFO2.3	L'utente autenticato deve poter tentare l'accesso premendo su un bottone di login	Capitolato
RFO2.4	Se l'utente non autenticato inserisce una username non esistente e preme sul bottone di login, deve essere visualizzato un messaggio d'errore di username inesistente	UC1.2
RFO2.5	Se l'utente non autenticato inserisce una password non corrispondente alla username inserita e preme sul bottone di login, deve essere visualizzato un messaggio d'errore di password errata	UC1.2
RFO2.6	Se l'utente non autenticato preme sul bottone di login senza aver inserito le credenziali di accesso, deve essere visualizzato un messaggio che inviti l'utente ad inserire delle credenziali	UC1.2
RFO2.7	Se il server non risponde in fase di login, deve essere visualizzato un messaggio d'errore che notifichi l'utente autenticato che il server è down	Stagista
RFO3	Se l'utente non autenticato riesce ad accedere al sistema, deve essere visualizzata la home page dell'applicazione	Capitolato
RFO3.1	L'utente autenticato deve visualizzare sulla home page la propria ragione sociale	Capitolato

Id Requisito	Descrizione	Fonti
RFO3.2	L'utente autenticato deve poter visualizzare sulla home page il tutorial dell'applicazione	UC2.2
RFO3.3	L'utente autenticato deve poter effettuare il logout dall'applicazione premendo su un bottone	UC2.1
RFO3.4	L'utente autenticato deve poter visualizzare sulla home page i propri articoli nel carrello	Capitolato
RFO3.4.1	Se l'utente autenticato non presenta articoli in carrello, deve essere visualizzato un messaggio che notifichi l'utente che il carrello è vuoto	Stagista
RFO3.4.2	La visualizzazione di un articolo in carrello deve comprendere una check-box per la selezione dell'articolo	Capitolato
RFO3.4.3	La visualizzazione di un articolo in carrello deve comprendere la quantità dell'articolo	Capitolato
RFO3.4.4	La visualizzazione di un articolo in carrello deve comprendere il codice dell'articolo	Capitolato
RFO3.4.5	La visualizzazione di un articolo in carrello deve comprendere la descrizione dell'articolo	Capitolato
RFO3.5	L'utente autenticato deve poter aggiungere un nuovo articolo al carrello	Capitolato UC2.3.1
RFO3.5.1	Per aggiungere un nuovo articolo al carrello, l'utente autenticato deve inserire un codice articolo	Capitolato UC2.3.1.1
RFO3.5.2	L'utente autenticato deve poter decidere se inserire il codice articolo con la scansione di un codice a barre o manualmente	Capitolato UC2.3.1.1
RFO3.5.3	Se l'utente autenticato decide di inserire il codice articolo con la scansione di un codice a barre, l'applicazione deve accedere alla fotocamera e permettere la cattura del codice a barre	Capitolato UC2.3.1.3
RFO3.5.3.1	L'applicazione deve permettere la cattura di un codice a barre di tipo EAN13	Capitolato

Id Requisito	Descrizione	Fonti
RFO3.5.3.2	L'applicazione deve permettere la cattura di un codice a barre di tipo EAN8	Capitolato
RFO3.5.3.3	L'applicazione deve permettere la cattura di un codice a barre di tipo CODE39	Capitolato
RFO3.5.3.4	L'applicazione deve permettere la cattura di un codice a barre di tipo UPC	Capitolato
RFO3.5.3.5	L'applicazione deve permettere la cattura di un codice a barre di tipo QR	Capitolato
RFO3.5.3.6	L'utente autenticato deve poter annullare la scansione del codice a barre con il tasto indietro del sistema operativo	Stagista
RFO3.5.3.7	Se l'utente autenticato annulla la scansione del codice a barre, deve essere visualizzato un messaggio di annullamento della scansione da parte dell'utente	UC2.3.2
RFO3.5.3.8	Se la scansione del codice a barre fallisce, la fotocamera deve essere chiusa e deve essere visualizzato un messaggio d'errore di scansione fallita	UC2.3.2
RFO3.5.3.9	Se la scansione del codice a barre va a buon fine ma il codice ottenuto non corrisponde ad un articolo venduto dall'azienda, deve essere visualizzato un messaggio d'errore che notifichi l'utente autenticato del problema	UC2.3.2
RFO3.5.3.10	Se la scansione del codice a barre va a buon fine e il codice ottenuto corrisponde ad un articolo venduto dall'azienda ma l'articolo è già presente in carrello, deve essere visualizzato un messaggio che inviti l'utente autenticato a cambiare la quantità dell'articolo in carrello	UC2.3.2
RFO3.5.3.11	Se la scansione del codice a barre va a buon fine e il codice ottenuto corrisponde ad un articolo venduto dall'azienda, deve essere visualizzata la pagina di aggiunta articolo	Capitolato

Id Requisito	Descrizione	Fonti
RFO3.5.4	Se l'utente autenticato decide di inserire il codice articolo manualmente, deve essere visualizzata una text box per l'inserimento del codice	Capitolato UC2.3.1.2
RFO3.5.4.1	Se l'utente autenticato inserisce un codice articolo che non corrisponde ad un articolo venduto dall'azienda, deve essere visualizzato un messaggio d'errore che notifichi l'utente autenticato del problema	UC2.3.2
RFO3.5.4.2	Se l'utente autenticato inserisce un codice articolo che corrisponde ad un articolo venduto dall'azienda ma l'articolo è già presente in carrello, deve essere visualizzato un messaggio che inviti l'utente autenticato a cambiare la quantità dell'articolo in carrello	UC2.3.2
RFO3.5.4.3	Se l'utente autenticato inserisce un codice articolo che corrisponde ad un articolo venduto dall'azienda, deve essere visualizzata la pagina di aggiunta articolo	Capitolato
RFO3.5.5	Nella pagina di aggiunta articolo, l'utente autenticato deve poter visualizzare il codice articolo	Capitolato
RFO3.5.6	Nella pagina di aggiunta articolo, l'utente autenticato deve poter visualizzare la descrizione dell'articolo	Capitolato
RFO3.5.7	Nella pagina di aggiunta articolo, l'utente autenticato deve poter visualizzare le informazioni dell'articolo	Capitolato
RFO3.5.8	La pagina di aggiunta articolo deve proporre all'utente autenticato una quantità minima per l'articolo da ordinare	Capitolato
RFO3.5.9	Nella pagina di aggiunta articolo, l'utente autenticato deve poter inserire una quantità per l'articolo da ordinare	Capitolato UC2.3.1.4
RFO3.5.10	Nella pagina di aggiunta articolo, l'utente autenticato deve poter inserire delle note per l'articolo da ordine	Capitolato UC2.3.1.5

Id Requisito	Descrizione	Fonti
RFO3.5.11	Nella pagina di aggiunta articolo, l'utente autenticato deve poter annullare l'inserimento dell'articolo	Capitolato
RFO3.5.12	Nella pagina di aggiunta articolo, l'utente autenticato deve poter confermare l'inserimento dell'articolo	Capitolato
RFO3.5.13	Se l'utente autenticato conferma l'inserimento dell'articolo ed ha inserito una quantità errata (troppo bassa, troppo alta o che non soddisfa uno step impostato dall'azienda), deve essere visualizzato un messaggio d'errore di errata quantità inserita	UC2.3.2
RFO3.5.14	Se l'utente autenticato conferma l'inserimento dell'articolo e i dati inseriti sono corretti, deve essere visualizzata la home page dell'applicazione e il nuovo articolo deve essere presente in carrello	Capitolato
RFO3.5.15	Se l'utente autenticato annulla l'inserimento dell'articolo, deve essere visualizzata la home page dell'applicazione e il carrello non deve aver subito cambiamenti	Capitolato
RFO3.5.16	Se viene persa la connettività nella pagina di aggiunta articolo, deve essere visualizzato un messaggio che inviti l'utente a chiudere l'applicazione per connettività assente	Stagista
RFO3.6	L'utente autenticato deve poter selezionare un articolo in carrello premendo sulla relativa check-box	Capitolato UC2.3.5
RFO3.7	L'utente autenticato deve poter deselectare un articolo in carrello selezionato premendo sulla relativa check-box	Capitolato UC2.3.6
RFO3.8	L'utente autenticato deve poter selezionare tutti gli articoli in carrello premendo su un bottone di seleziona/deseleziona tutti presente sulla home page	Capitolato UC2.3.7

Id Requisito	Descrizione	Fonti
RFO3.9	L'utente autenticato deve poter deselectare tutti gli articoli in carrello premendo su un bottone di seleziona/deseleziona tutti presente sulla home page	Capitolato UC2.3.8
RFO3.10	L'utente autenticato deve poter modificare un articolo in carrello premendo sopra la visualizzazione dell'articolo sul carrello	Capitolato UC2.3.3
RFO3.10.1	Se l'utente autenticato preme sulla visualizzazione di un articolo in carrello, deve essere richiesta la conferma di modifica tramite la visualizzazione di un dialog	Stagista
RFO3.10.2	Se l'utente autenticato accetta di modificare l'articolo sul dialog di conferma, deve essere aperta la pagina di modifica articolo	Stagista
RFO3.10.3	Se l'utente autenticato rifiuta di modificare l'articolo sul dialog di conferma, deve essere visualizzata la home page senza cambiamenti sugli articoli in carrello	Stagista
RFO3.10.4	Nella pagina di modifica articolo, l'utente autenticato deve poter visualizzare il codice articolo dell'articolo selezionato	Capitolato
RFO3.10.5	Nella pagina di modifica articolo, l'utente autenticato deve poter visualizzare la descrizione dell'articolo selezionato	Capitolato
RFO3.10.6	Nella pagina di modifica articolo, l'utente autenticato deve poter visualizzare le informazioni dell'articolo selezionato	Capitolato
RFO3.10.7	La pagina di modifica articolo deve proporre all'utente autenticato la quantità precedentemente inserita per l'articolo selezionato	Capitolato
RFO3.10.8	Nella pagina di modifica articolo, l'utente autenticato deve poter cambiare la quantità per l'articolo selezionato	Capitolato UC2.3.3.1
RFO3.10.9	La pagina di modifica articolo deve proporre all'utente autenticato le note precedentemente inserite per l'articolo selezionato	Capitolato

Id Requisito	Descrizione	Fonti
RFO3.10.10	Nella pagina di modifica articolo, l'utente autenticato deve poter modificare le note per l'articolo selezionato	Capitolato UC2.3.3.1
RFO3.10.11	Nella pagina di modifica articolo, l'utente autenticato deve poter annullare la modifica dell'articolo	Capitolato
RFO3.10.12	Nella pagina di modifica articolo, l'utente autenticato deve poter confermare la modifica dell'articolo	Capitolato
RFO3.10.13	Se l'utente autenticato conferma la modifica dell'articolo ed ha inserito una quantità errata (troppo bassa, troppo alta o che non soddisfa uno step impostato dall'azienda), deve essere visualizzato un messaggio d'errore di errata quantità inserita	UC2.3.4
RFO3.10.14	Se l'utente autenticato conferma la modifica dell'articolo e i dati modificati sono corretti, deve essere visualizzata la home page dell'applicazione e l'articolo modificato deve essere presente in carrello con le modifiche apportate	Capitolato
RFO3.10.15	Se l'utente autenticato annulla la modifica dell'articolo, deve essere visualizzata la home page dell'applicazione e l'articolo selezionato non deve essere stato modificato	Capitolato
RFO3.10.16	Se viene persa la connettività nella pagina di modifica, deve essere visualizzato un messaggio che inviti l'utente a chiudere l'applicazione per connettività assente	Stagista
RFO3.11	L'utente autenticato deve poter rimuovere gli articoli selezionati sul carrello premendo su un bottone presente sulla home page	Capitolato UC2.3.9
RFO3.11.1	Se l'utente autenticato preme sul bottone di rimozione articoli senza aver selezionato alcun articolo, deve essere visualizzato un messaggio che inviti l'utente a selezionare degli articoli	UC2.3.11

Id Requisito	Descrizione	Fonti
RFO3.11.2	Se l'utente autenticato preme sul bottone di rimozione articoli avendo selezionato degli articoli sul carrello, deve essere richiesta la conferma di eliminazione tramite la visualizzazione di un dialog	Stagista
RFO3.11.3	Se l'utente autenticato conferma l'eliminazione sul dialog di conferma, gli articoli precedentemente selezionati devono essere rimossi dal carrello	Capitolato
RFO3.11.4	Se l'utente autenticato rifiuta l'eliminazione sul dialog di conferma, il carrello non deve subire cambiamenti	Capitolato
RFO3.12	L'utente autenticato deve poter inviare un ordine composto dagli articoli selezionati sul carrello premendo su un bottone presente sulla home page	Capitolato UC2.3.10
RFO3.12.1	Se l'utente autenticato preme sul bottone di invio ordine senza aver selezionato alcun articolo, deve essere visualizzato un messaggio che inviti l'utente a selezionare degli articoli	UC2.3.11
RFO3.12.2	Se l'utente autenticato preme sul bottone di invio ordine avendo selezionato degli articoli sul carrello, deve essere visualizzato il modal di invio ordine	Stagista
RFO3.12.3	Nel modal di invio ordine, l'utente autenticato deve poter visualizzare il codice cliente	Capitolato
RFO3.12.4	Nel modal di invio ordine, l'utente autenticato deve poter visualizzare la descrizione cliente	Capitolato
RFO3.12.5	Nel modal di invio ordine, l'utente autenticato deve poter visualizzare il codice del documento da generare per l'ordine	Capitolato
RFO3.12.6	Nel modal di invio ordine, l'utente autenticato deve poter visualizzare la descrizione del documento da generare per l'ordine	Capitolato

Id Requisito	Descrizione	Fonti
RFO3.12.7	Il modal di invio ordine deve proporre all'utente autenticato la data corrente come data d'ordine	Capitolato
RFO3.12.8	Nel modal di invio ordine, l'utente autenticato deve poter modificare la data d'ordine proposta	Capitolato UC2.3.10.1
RFO3.12.9	Nel modal di invio ordine, l'utente autenticato deve poter inserire delle note per l'ordine	Capitolato UC2.3.10.2
RFO3.12.10	Nel modal di invio ordine, l'utente autenticato deve poter confermare l'invio dell'ordine	Capitolato
RFO3.12.11	Nel modal di invio ordine, l'utente autenticato deve poter annullare l'invio dell'ordine	Capitolato
RFO3.12.12	Se l'utente autenticato decide di annullare l'invio dell'ordine, il modal di invio ordine deve essere chiuso	Stagista
RFO3.12.13	Se l'utente autenticato decide di confermare l'invio dell'ordine, deve essere richiesta la conferma di invio ordine tramite un dialog	Stagista
RFO3.12.14	Se l'utente autenticato conferma l'invio dell'ordine nel dialog di conferma e l'ordine è andato a buon fine, deve essere visualizzato un messaggio di ordine inviato con successo e una mail di conferma deve essere inviata sia all'utente che all'azienda	Capitolato
RFO3.12.14.1	La mail inviata dal sistema al momento dell'invio dell'ordine deve contenere in oggetto la data di registrazione dell'ordine	Capitolato
RFO3.12.14.2	La mail inviata dal sistema al momento dell'invio dell'ordine deve contenere in oggetto il codice azienda dell'utente che ha effettuato l'ordine	Capitolato
RFO3.12.14.3	La mail inviata dal sistema al momento dell'invio dell'ordine deve contenere in oggetto la tipologia di documento generato con la registrazione dell'ordine	Capitolato

Id Requisito	Descrizione	Fonti
RFO3.12.14.4	La mail inviata dal sistema al momento dell'invio dell'ordine deve contenere nel corpo il codice cliente del cliente che ha effettuato l'ordine	Capitolato
RFO3.12.14.5	La mail inviata dal sistema al momento dell'invio dell'ordine deve contenere nel corpo una tabella contenente i dati degli articoli ordinati	Capitolato
RFO3.12.14.6	Per ogni articolo in tabella deve essere indicata la quantità ordinata	Capitolato
RFO3.12.14.7	Per ogni articolo in tabella deve essere indicato il codice articolo	Capitolato
RFO3.12.14.8	Per ogni articolo in tabella deve essere indicata la descrizione dell'articolo	Capitolato
RFO3.12.14.9	La mail inviata dal sistema al momento dell'invio dell'ordine deve contenere nel corpo l'avviso di non rispondere alla mail	Capitolato
RFO3.12.15	Se l'utente autenticato conferma l'invio dell'ordine nel dialog di conferma e l'ordine non è andato a buon fine, deve essere visualizzato un messaggio d'errore di mancato invio dell'ordine	Capitolato
RFO3.12.16	Se l'utente autenticato annulla l'invio dell'ordine nel dialog di conferma, il dialog di conferma deve essere chiuso	Stagista
RFO3.13	Se viene persa la connettività nella home page, deve essere visualizzato un messaggio che inviti l'utente a chiudere l'applicazione per connettività assente	Stagista

Tabella 4.1: Tabella del tracciamento dei requisiti funzionali

4.2.2 Requisiti qualitativi

Id Requisito	Descrizione	Fonti
RQO1	L'applicazione deve essere fornita di manuale utente	Tutor
RQD2	L'applicazione deve essere fornita di manuale sviluppatore	Tutor

Tabella 4.2: Tabella del tracciamento dei requisiti qualitativi

4.2.3 Requisiti di vincolo

Id Requisito	Descrizione	Fonti
RVO1	L'applicazione deve essere sviluppata con ambiente multiplatforma (preferibilmente PhoneGap o Xamarin)	Capitolato
RVO2	L'applicazione deve utilizzare il database standard di VisionENTERPRISE, senza modifica alla sua struttura	Capitolato
RVO3	L'applicazione deve interfacciarsi con la fotocamera per la lettura di codici a barre	Capitolato
RVO4	L'applicazione deve funzionare completamente online, nessun dato deve essere memorizzato in locale e nessun database deve essere sincronizzato	Capitolato
RVD5	L'applicazione deve utilizzare un database SQL Server lato web	Capitolato
RVD6	L'applicazione deve supportare codici a barre bidimensionali (QRcode)	Capitolato
RVF7	L'applicazione deve condividere i dati di login con l'applicazione <i>moviDOC_G</i>	Capitolato
RVF8	L'applicazione deve avere lo stesso stile di interfaccia di <i>moviDOC</i>	Capitolato
RVF9	L'applicazione deve permettere la firma digitale del cliente a fine ordine	Capitolato

Tabella 4.3: Tabella del tracciamento dei requisiti di vincolo

4.2.4 Riepilogo requisiti

I requisiti sono di seguito raggruppati in una tabella riepilogativa, in modo da comprendere la grandezza del progetto in termini di requisiti.

Tipologia	Obbligatorio	Desiderabile	Facoltativo	Totale
Funzionale	102	0	0	102
Qualitativo	1	1	0	2
Di vincolo	4	2	3	9
Totale	107	3	3	113

Tabella 4.4: Riepilogo requisiti

4.2.5 Validazione dei requisiti

Al termine dell'analisi dei requisiti, lo stagista ha validato i requisiti tramite delle interazioni con il tutor aziendale. Nello specifico ci si è assicurati che i requisiti individuati rappresentassero il sistema richiesto dal tutor aziendale. Lo stagista ha fatto in modo aderire allo standard IEEE 830-1998 per la scrittura dei requisiti. Lo standard IEEE 830-1998 individua otto qualità essenziali per i requisiti:

- * assenza di ambiguità;
- * correttezza;
- * completezza;
- * verificabilità;
- * consistenza;
- * modificabilità;
- * tracciabilità;
- * ordinamento per rilevanza.

Capitolo 5

Progettazione e codifica

Breve introduzione al capitolo

5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

5.2 Ciclo di vita del software

5.3 Progettazione

5.3.0.1 Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.4 Design Pattern utilizzati

5.5 Codifica

Capitolo 6

Verifica e validazione

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Convenzioni

In questo capitolo vengono presentate le convenzioni utilizzate per la classificazione di casi d'uso e requisiti.

A.1 Casi d'uso

Ogni caso d'uso è classificato tramite il seguente formalismo:

$$\text{UC}\{\text{codice_identificativo}\}$$

dove:

- * **codice_identificativo**: è un codice composto da una serie di numeri separati tramite punto che identificano il caso d'uso in maniera univoca e lo esprimono gerarchicamente.

A.2 Requisiti

Ogni requisito è classificato tramite il seguente formalismo:

$$\text{R}\{\text{X}\}\{\text{Y}\}\{\text{codice_identificativo}\}$$

dove:

- * **X** specifica la tipologia di requisito:
 - *F*: requisito funzionale;
 - *Q*: requisito qualitativo;
 - *V*: requisito di vincolo.
- * **Y** indica uno dei seguenti gradi di necessità:
 - *O*: requisito obbligatorio;
 - *D*: requisito desiderabile;
 - *F*: requisito facoltativo.
- * **codice_identificativo**: è un codice composto da una serie di numeri separati tramite punto che identificano il requisito in maniera univoca e lo esprimono gerarchicamente.

Appendice B

Appendice A

Citazione

Autore della citazione

Bibliografia