



Università degli studi di Napoli Parthenope

Progetto Ingegneria del Software e Interazione Uomo Macchina

Parthenope

System Design Document

Gruppo:

Riccardo Andrea Spinosa

0124002253

Alessandro massadoro

0124002450

Professori:

Mariacarla Staffa

Paola Barra



Premessa Importante:	2
1.0 Introduction	3
1.1 Proposed of System	3
1.2 Design Goals:	4
1.3 References	4
2.0 diagramma delle classi	7
3.0 Proposed software architecture	10
4.0 Subsystem decomposition	11
5.0 Hardware/Software Mapping	13
6.0 Persistent data Management	14
7.0 Sicurezza:	16
7.1 Attacchi tipici	16
7.2 Soluzioni:	17
8.0 Boundary Conditions	18
9.0 Glossario	19

Premessa Importante:

Per una migliore leggibilità dei diagrammi e del progetto, sono stati omessi alcuni casi d'uso o funzionalità che non riguardano direttamente l'idea del progetto.

Ad esempio password dimenticata è una procedura scontata all'interno del login che non è stata appositamente aggiunta.

1.0 Introduction

1.1 Proposed of System

L'obiettivo del sistema vuole essere la realizzazione di una dashboard che permetta di gestire e aggiungere annunci attraverso un'interfaccia intuitiva. L'utente non registrato che si interfaccia la prima volta al sistema dovrà eseguire obbligatoriamente la registrazione (email,password,partita iva, etc.) .

Si premette che Alcuni utenti hanno la partita iva, e possono eseguire operazioni aggiuntive su piattaforme specifiche che la richiedono (amazon, wish...) altri no e dunque potranno interfacciarsi solo a piattaforme che non la richiedono.

Gli annunci, ripartiti in piattaforme diverse, sono gestiti direttamente dalla nostra app. Collezionano tutti gli annunci inseriti nelle varie numerose piattaforme. Una volta registrato l'utente potrà effettuare il login e interfacciarsi al sistema. Nello specifico, la dashboard reindirizza l'utente alla home e mostra tutti gli annunci attivi delle varie piattaforme. La dashboard con apposita sezione permette di aggiungere nuovi annunci, composti da foto del prodotto selezionando le piattaforme di pubblicazione e altre info necessarie alla sua pubblicazione. In maniera intuitiva sarà possibile effettuare dall'applicazione eventuali modifiche dell'inserzione o la sua eliminazione. La sezione cerca, permetterà di trovare in maniera rapida un annuncio.

Ogni annuncio invia periodicamente in maniera real-time notifiche all'utente per segnalare eventi quali vendita del prodotto con specifica piattaforma, o messaggi con apposita sezione segnalando la specifica piattaforma sulla quale è stato ricevuto.

La dashboard offrirà la sezione profilo, dove potranno essere visualizzate varie informazioni e sarà possibile in tale sezione scaricare la fattura o visualizzare il wallet- Il wallet conterrà solo informazioni riguardanti le transazioni degli articoli venduti.(+50€ -> amazon)

Il tutto verrà gestito tramite APIs offerte dalle varie piattaforme.

esempio APIs offerte: <https://developer.ebay.com/develop/apis>

1.2 Design Goals:

- Usabilità:
La dashboard deve avere un'interfaccia chiara e semplice da utilizzare.
- Conformità alle linee guida:
L'interfaccia deve seguire le linee guida del Interface Guidelines di Apple.
- Piattaforma di destinazione:
Dispositivi ios.
- Throughput:
Il sistema deve essere in grado di immagazzinare e processare almeno 10.000 prodotti nei vari market e le eventuali notifiche real-time.
- Disponibilità:
Il sistema è sempre accessibile dagli utenti e le funzionalità possono sempre essere eseguite.

1.3 References

Api, ogni volta che si parla di API facciamo riferimento a servizi di terzi offerti quindi dalle varie piattaforme su cui vengono inseriti annunci, facciamo riferimento alle API offerte da Ebay: <https://developer.ebay.com/develop/apis>

Come

4.0 Abbott's objects identification

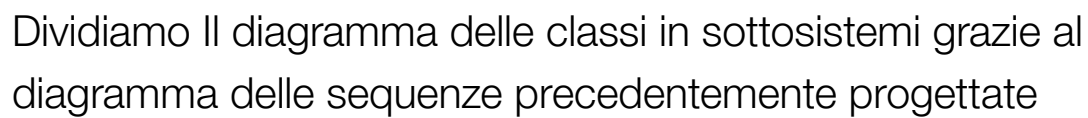
Entity	
Utente Registrato	Utente Generico
Utente Registrato Con Partita Iva	Utente con funzionalità in più
Piattaforma	Piattaforma con cui l'applicazione necessita di comunicare per poter fornire i servizi implementati
Annuncio	contiene tutti i dati relativi agli annunci

Boundary	
Login	Modulo dove inserire i dati per accedere
Registrazione	Modulo per registrarsi alla Dashboard
Vendi	Schermata Che porta alla schermata dove è possibile aggiungere gli articoli
(+) Carica Articolo	Pulsante che permette di aggiungere un articolo
Piattaforma	Spunta che permette di scegliere le piattaforme dove pubblicare gli articoli
Home	Pulsante che permette di navigare alla schermata dove vengono visualizzati tutti gli articoli messi in vendita
Barra di ricerca	Barra presente nella schermata home che permette di trovare articoli o messaggi.
Notifica	Schermata in cui si visualizza la notifica ricevuta.
Messaggio	Messaggio inviato o ricevuto dall'utente
Modifica	Pulsante che permette di modificare l'annuncio
Elimina	Pulsante che permette di eliminare un annuncio
Scarica	Pulsante che permette di scaricare la fattura
Profilo	Schermata con i dati dell'utente
Wallet	Pulsante disponibile dalla schermata profilo elenca tutti gli eventuali guadagni, mostrando la piattaforma da cui derivano

Controll	
Login()	Permette l'accesso al sistema

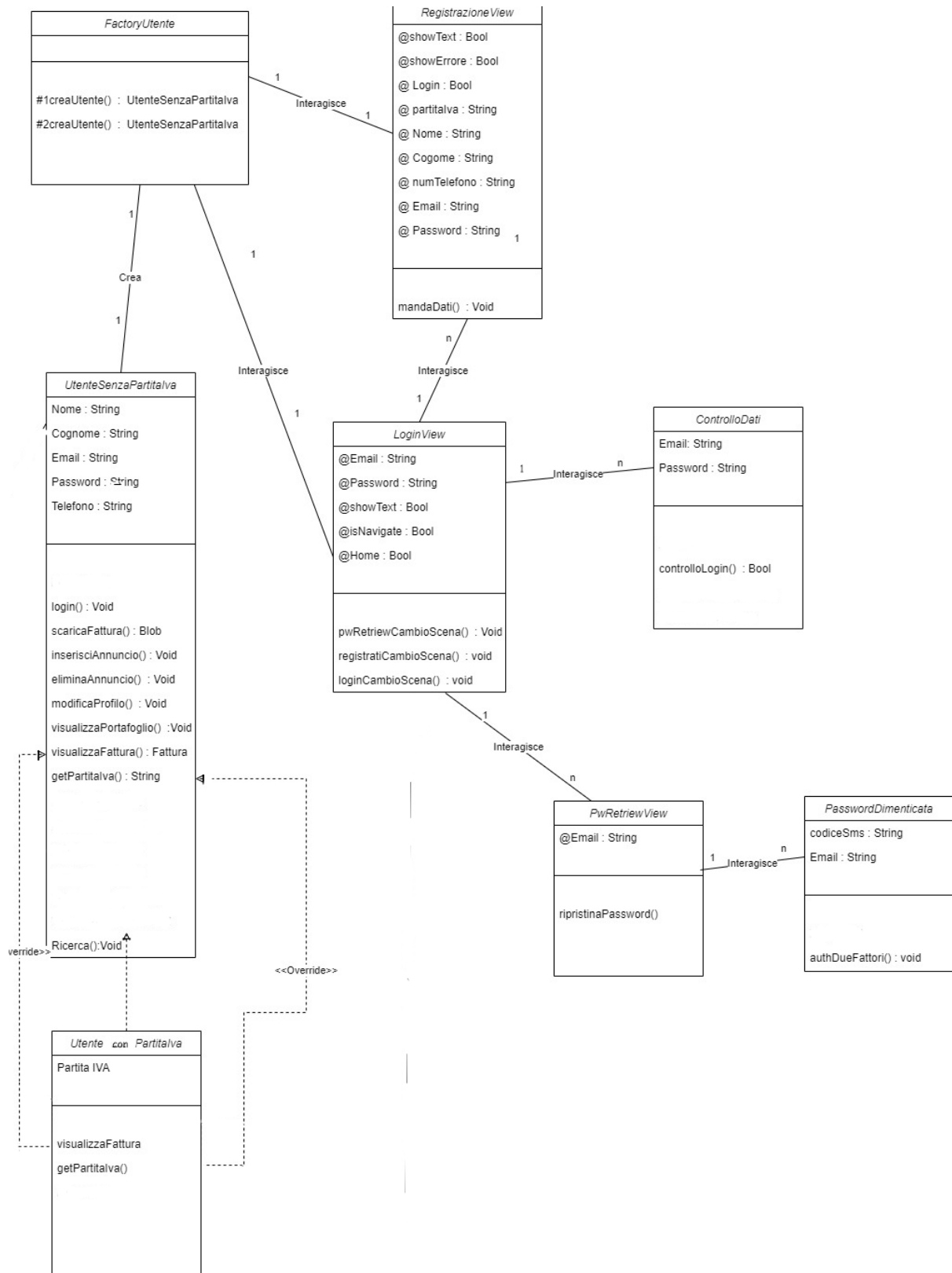
Controll	
Registrazione()	Permette la registrazione
inserisciAnnuncio()	Inserisce un annuncio di vendita
eliminaAnnuncio()	Elimina un annuncio
modificaAnnuncio()	Modifica un annuncio come ad esempio il suo titolo o prezzo.
modificaProfilo()	Permette di modificare i dati di un utente
visualizzaPortafoglio()	Visualizza il portafoglio virtuale
inviaMessaggio()	Permette di inviare un messaggio
riceveMessaggio()	Resta in ascolto di messaggi in arrivo
scaricaFattura()	Scarica la fattura di un articolo venduto attraverso l'applicazione.
visualizzaFattura()	permette di visualizzare le fatture
ricerca()	permette di filtrare gli annunci
ripristinaPassword()	permette il ripristino di una password dimenticata
controlloLogin()	controlla che i dati di accesso siano corretti
creaUtente()	metodo factory che permette la creazione di un utente run time a seconda che abbia oppure no la partita iva
inserisciTransazione()	inserisce i dati di una transazione avvenuta nel file system
inserisciFattura()	inserisce i dati di una fattura nel file system
private init()	meccanismo di protezione al singleton per non farlo istanziare

2.0 diagramma delle classi



Dividiamo il diagramma delle classi in sottosistemi grazie al diagramma delle sequenze precedentemente progettate

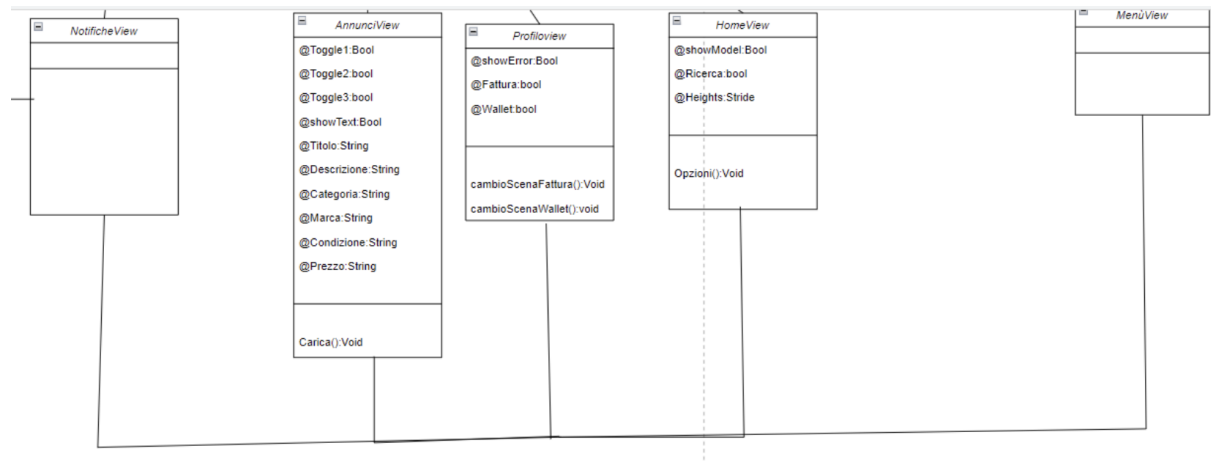
Sotto sistema di autenticazione:



Il sotto sistema di autenticazione permette all'utente di poter accedere al sistema e di essere autenticato. Nel caso in cui non sia registrato dovrà prima registrarsi così da

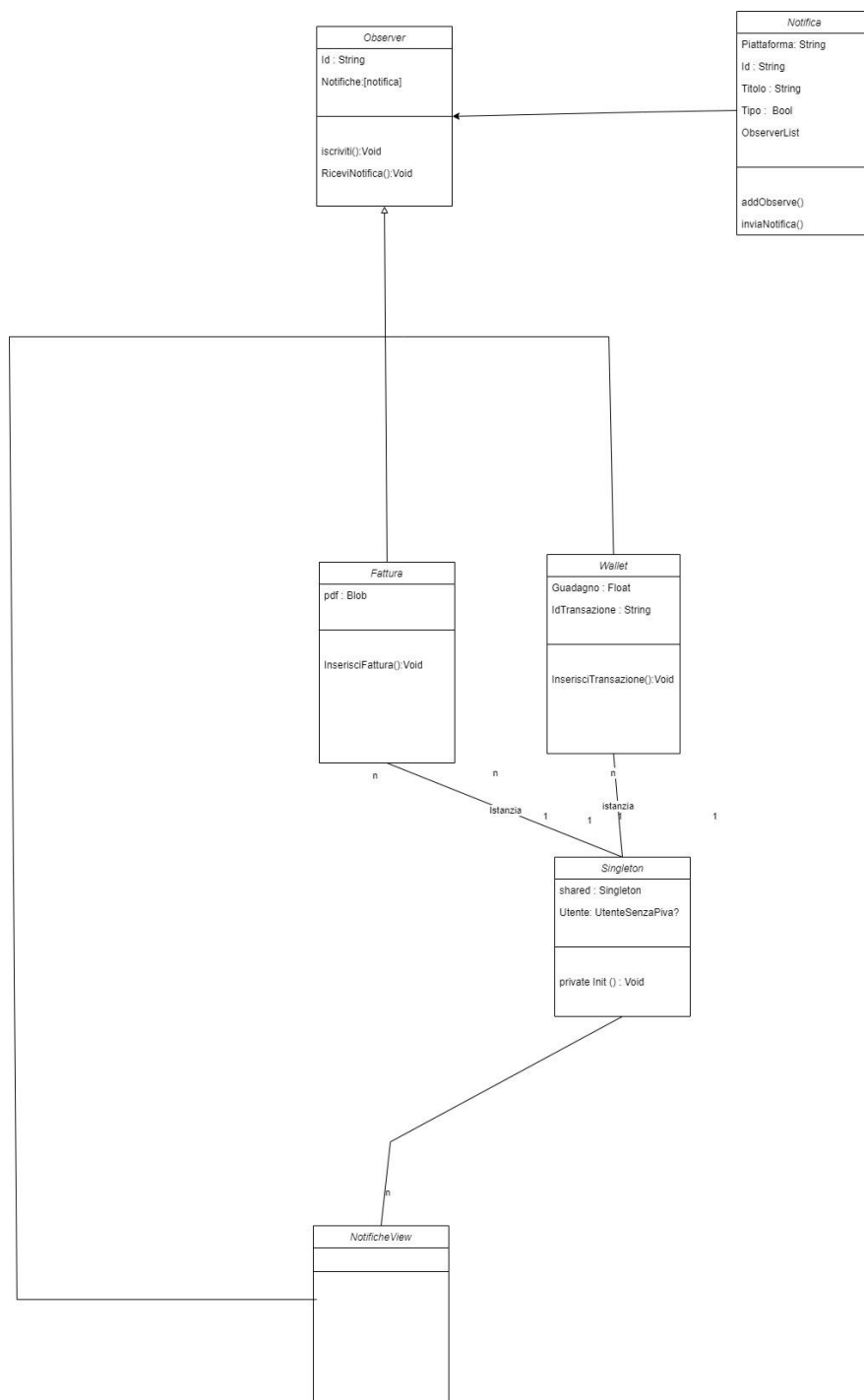
procedere all'autenticazione mentre nel caso l'utente non ricordi la propria password permette il recupero di quest'ultima così da permettere nuovamente l'autenticazione.

Sotto sistema Menu:



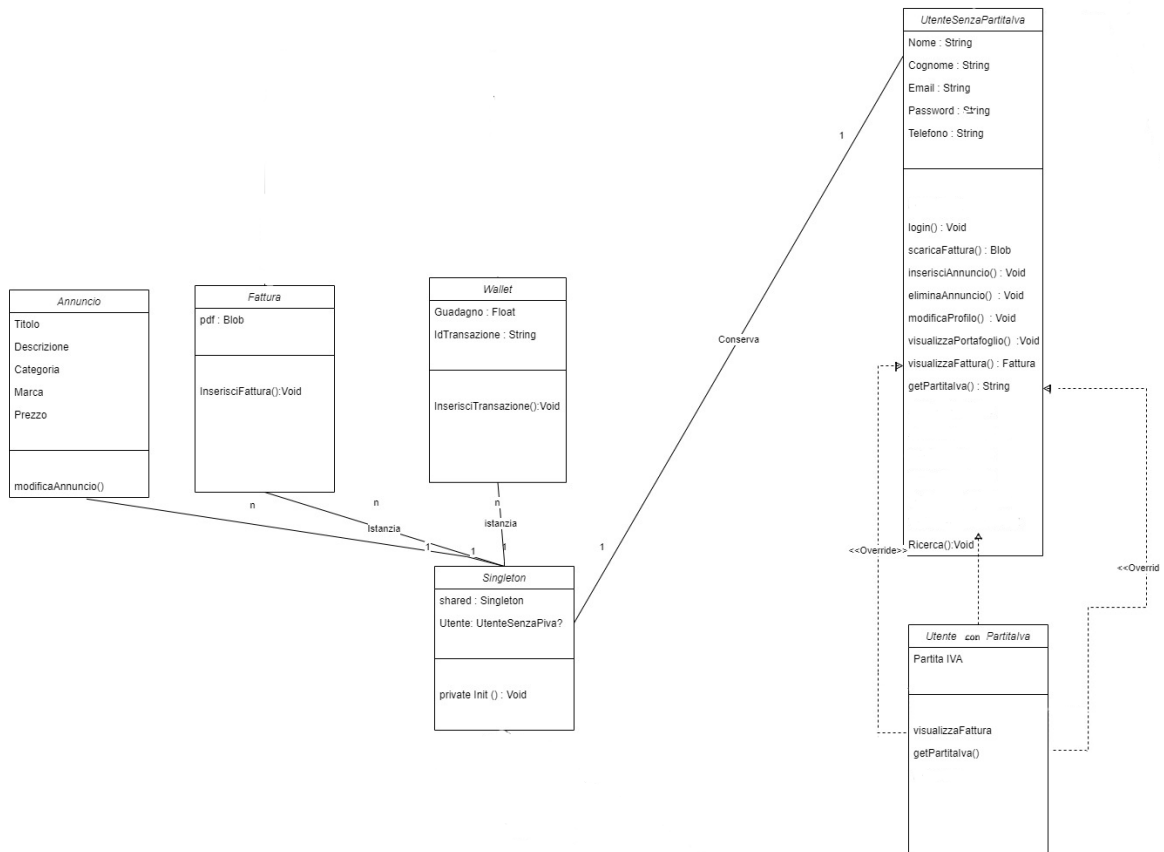
Il sotto sistema gestisce la gestione delle viste del menù e permette lo spostamento, così da rendere più chiaro il mondo in cui l'utente può spostarsi all'interno della nostra applicazione.

Sotto sistema notifiche in tempo reale



Tale sottosistema rende chiaro come il sistema riceve le notifiche in tempo reale e come aggiorna le sue dipendenze una volta arrivate.

Sotto sistema gestione operazioni utente loggato



In questo sotto sistema analizziamo la componente di gestione delle operazioni dell'utente loggato, troviamo le classi che consentono la gestione di quest'ultimo nel dettaglio. Queste classi permettono all'utente di eseguire tutte le operazioni permesse dal sistema senza considerare le viste dove vengono mostrati i risultati ma solo la logica di modellazione.

3.0 Proposed software architecture

Lo stile architetturale scelto è client/server.

Esistono due tipi di client:

- L'utente Registrato, si collega all'applicazione accedendo alle funzionalità del sistema
- L'utente registrato con partita iva che utilizza il sistema sempre come utente registrato solo che sblocca funzionalità in più.
- L'utente non registrato ha come unica opzione la registrazione.

Il server utilizza delle API di terze parti messe a disposizione stesso dalle piattaforme che consentono la gestione dei vari annunci che l'utente intende inserire nella piattaforma di appartenenza per fornire i servizi ai client.

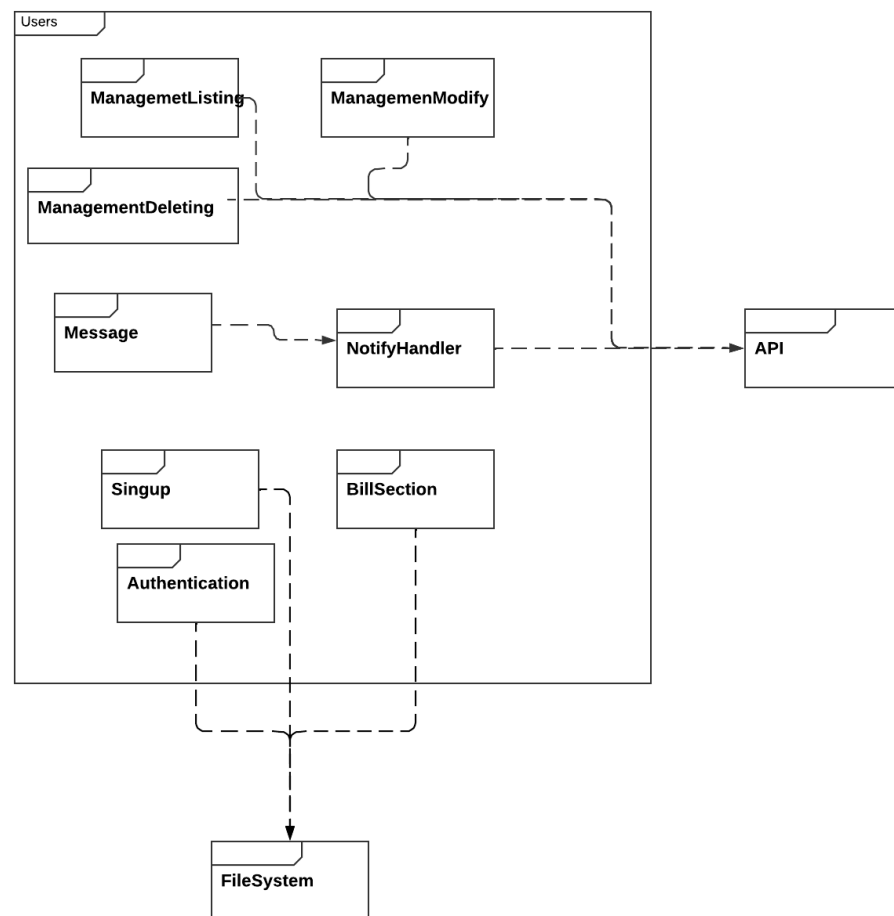
Per citare un esempio Ebay offre Sell APIs.

Il sistema utilizza come supporto dati un File System

4.0 Subsystem decomposition

Abbiamo individuato nel nostro sistema alcuni sottosistemi fondamentali, che ci aiutano a suddividere il sistema principale in parti più piccole aumentando quindi la facilità di gestione di quest'ultime.

Utilizziamo questa suddivisione per migliorare la modularità, la complessità, ottenendo una manutenzione ottimale e facilitando l'evoluzione del nostro sistema in futuro.



Authentication: è il modulo che si occupa dell'autenticazione dell'utente, cioè gestisce il login degli utenti al sistema.

Sign Up: è il modulo che gestisce la registrazione dei nuovi utenti che effettuano l'accesso per la prima volta

Message: è l'area che gestisce la messaggistica di risposta che il sistema offre per permettere agli utenti di rispondere a messaggi ricevuti sulle piattaforme su cui sono posizionati gli annunci.

Bill Section: permette la gestione delle fatture per gli utenti con partita iva.

Notify Handler: è l'area del sistema che riceve le notifiche dai server API delle piattaforme, ci permette di isolare la ricezione di notifiche da parte dei servizi di terzi date dalle piattaforme, inoltre notifica gli utenti delle interazioni di vendita o d'interazione con i loro annunci sulle piattaforme.

Management Listing: area dedicata all'inserimento di nuovi annunci da parte degli utenti, comunica con le piattaforme su cui è necessario eseguire l'operazione.

Management Deleting: permette l'eliminazione dei vecchi annunci da parte degli utenti, comunica con le piattaforme su cui è necessario eseguire l'operazione.

Management Modify: permette la modifica degli annunci, comunica con le piattaforme su cui è necessario eseguire l'aggiornamento.

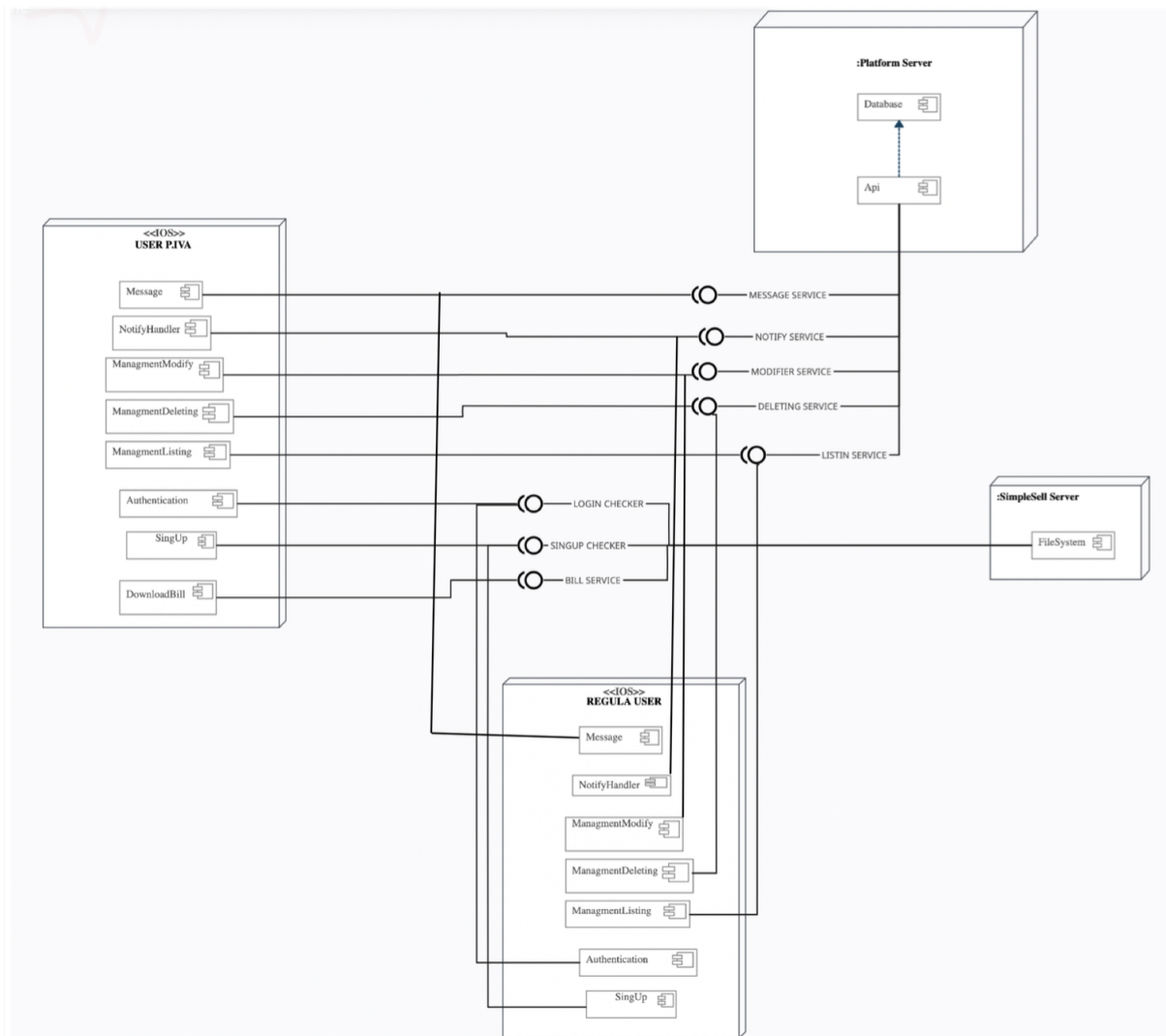
API: sottosistema facente da ponte tra i client ed il server, i sono i servizi offerti dalle piattaforme di vendita, che ci permettono di gestire dalla nostra applicazione tutte le operazioni inerenti agli annunci e alla gestione dell'utente.

File System: Contiene le informazioni riguardo le fatture e gli utenti registrati

5.0 Hardware/Software Mapping

Il sistema attualmente è pensato per essere utilizzato solo da smartphone, più precisamente dispositivi IOS, risulta quindi essere un vincolo della nostra progettazione attuale.

Troviamo due differenti implementazioni, per gli utenti professionisti che vendono su piattaforme utilizzando anche partita iva e utenti non professionisti i quali vendono pochi articoli l'anno.



6.0 Persistent data Management

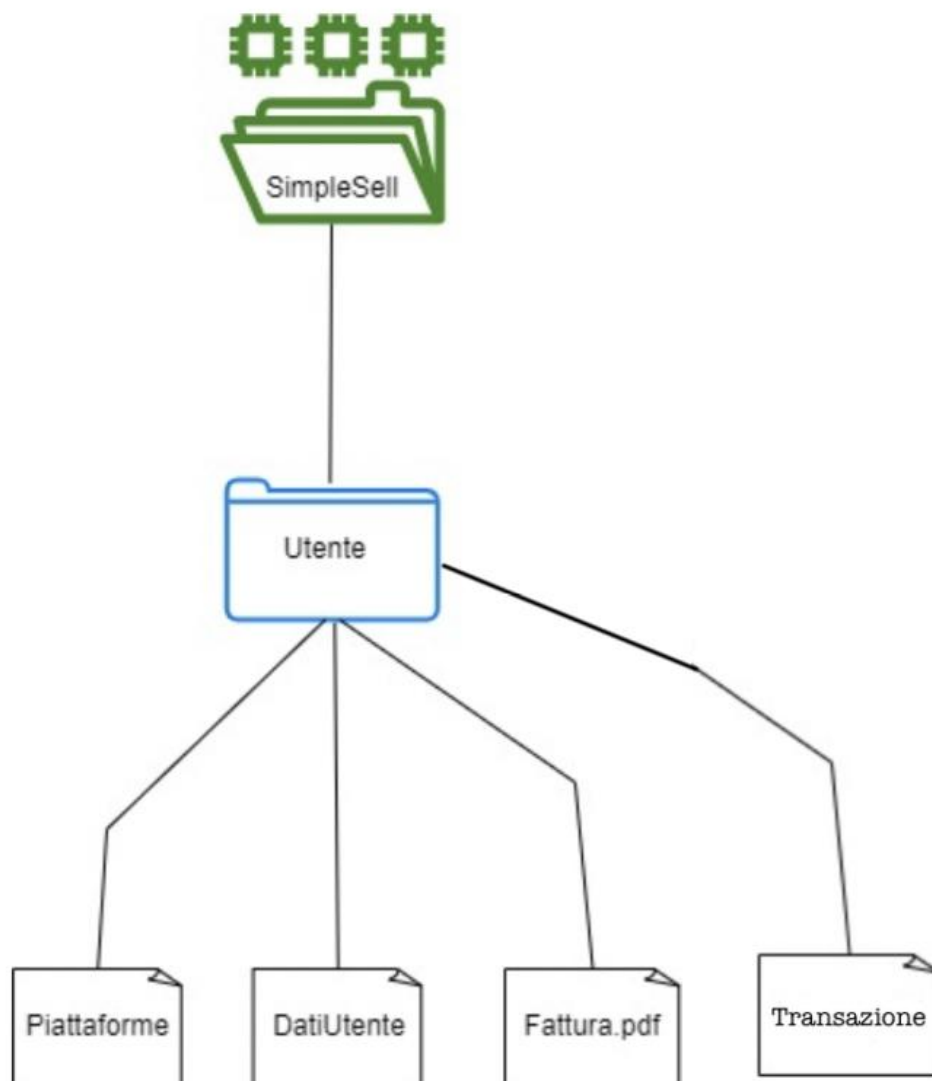
I dati persistenti del sistema di SimpleSells sono memorizzati all'interno di un File System. La scelta del File System è scaturita essenzialmente da motivi di costo. L'utilizzo di un database necessita di ulteriori costi di progettazione oltre che di implementazione, inoltre prolunga di molto l'uscita del prodotto finale. Un database necessita di manutenzione e di essere attivi h24 online. Il sistema per implementare i servizi usufruisce pressoché di servizi esterni attraverso API's di terzi, tali dati dunque non necessitano di essere memorizzati persistentemente perchè messi a disposizione dalle piattaforme esterne utilizzate. I dati memorizzati sono solo Dati utenti, le piattaforme su cui opera e le fatture in formato pdf, essendo una struttura molto

semplice, l'utilizzo di un database comporterebbe solo costi senza reali motivi di utilizzo.

Constatata la mancanza di problemi di spazio per la memorizzazione di dati persistenti si è scelto l'utilizzo di un fileSystem in quanto trattiamo dati misti, sia dati strutturati che non dati strutturati. Un esempio di dati necessari alla memorizzazione persistente corrispondono a quelli in fase di registrazione dell'utente oltre alle fatture in formato pdf. Motivato l'utilizzo di un filesystem sorge un problema di sicurezza. Ogni attore ha due chiavi: una per cifrare, definita chiave pubblica o pubkey, una per decifrare, definita chiave privata. Quindi, tornando ai due attori Riccardo e Alessandro, Alessandro userà la chiave pubblica di Riccardo per cifrare il messaggio; Riccardo userà la sua chiave privata per decifrarlo.

Teniamo conto che per quanto riguarda l'operazione di registrazione di un nuovo utente dovrà essere eseguita un'operazione di scrittura, quando si eseguono delle operazioni di scrittura le operazioni vanno incontro a problemi di race condition, si necessita dunque di sincronizzare le operazioni in mutua esclusione. I processi di lettura invece non hanno bisogno di essere sincronizzati, questo è un chiaro esempio di uno dei classici problemi di sincronizzazione "lettori e scrittori".

Successivamente segue lo schema uml e la progettazione del filesystem:



All'interno della cartella SimpleSell sono contenute directory per un numero pari al numero di utenti presenti nell'app: ogni utente ha la directory personale e facilmente identificabile in quanto il nome della directory coincide con il suo identificativo all'interno dell'applicazione. Salviamo le fatture in quanto nonostante sono accessibili da APIs come il resto dei dati diamo all'utente questa funzionalità aggiuntiva così che possa visualizzare e lavorare sulle fatture anche senza che i servizi delle piattaforme siano funzionanti, lo stesso non sarebbe possibile con gli altri elementi che necessitano obbligatoriamente della piattaforma di riferimento. Oltre a salvare i Dati Utente necessari per la registrazione in piattaforme abbiamo i dati relative alle varie piattaforme che si necessitano per usufruire delle APIs. Inoltre la transazioni sono le transazioni registrate all'interno del wallet, per rappresentare tutti i pagamenti ricevuti.

Si noti come lo schema del filesystem sia molto semplice.

7.0 Sicurezza:

L'app potrà essere utilizzata solo da utenti registrati dunque il sistema necessita di garantire la sicurezza dei loro dati ed evitare accessi non autorizzati. Si noti che la nostra app permette solo di gestire gli annunci e anche in caso di furto di dati non saranno possibili furti di denaro o acquisti non autorizzati in quanto il sistema non permette la raccolta di tali informazioni e né li gestisce. Tuttavia il sistema necessita di misure di sicurezza in quanto la perdita del proprio account potrebbe risultare in un danno consistente. (Eventuali negozi professionali che vendono grandi quantità di prodotti.)

7.1 Attacchi tipici

Tipici attacchi utilizzati per il furto di dati sensibili sono:

1. Utilizzo di password non sicure da parte dei nostri clienti, Gli hacker spesso posseggono una lista contenente tutte le password più comuni e con un semplice algoritmo di confronto tentano di accedere con una di quelle password, spesso ci riescono, perché tali password sono molto comuni e facili da ricordare.
2. Brute-Force se le password non sono forti ma semplici un hacker potrebbe utilizzare un algoritmo brute force per cercare di forzare la combinazione utilizzata come password per accedere all'applicazione, ciò è possibile con l'utilizzo di password corte e poco forti, se una password è composta da una sola semplice parola mettiamo caso maradona un semplice algoritmo brute-force che tenta combinazioni composte da parole, anche se la password non risulta molto corta facilmente sarebbe reperita dal malintenzionato accedendo così all'applicazione e carpendo dati sensibili.
3. Password contenente il proprio nome, spesso, gli utenti per potersi ricordare facilmente la password utilizzano il proprio nome all'interno della password in combinazione magari con la propria data di nascita, un hacker che magari ha carpito tali informazioni attraverso il profilo pubblico su facebook o instagram potrebbe tentare

di ricostruire la password così ottenendo l'accesso all'applicazione, è una variante del brute-force con recupero d'informazioni personali da parte del malintenzionato.

4. Attacchi keylogger, quelli maggiormente utilizzati e facili da implementare. Gli hacker attraverso un link reindirizza il malcapitato scaricando all'interno del proprio computer un programma keylogger che semplicemente legge da tastiera ciò che noi digitiamo e magari a quali richieste http siamo connessi, successivamente attraverso il terminale kali-linux controllano ciò che noi digitiamo carpendo informazioni personali come email e password.

5. Intercettazione dei messaggi per carpire informazioni importanti, quelli meno utilizzati perché più difficili da eseguire, un malintenzionato potrebbe intercettare un codice oppure i messaggi inviati semplicemente intercettando con un apposito dispositivo che riceve i messaggi al posto nostro. Tali informazioni potrebbero essere utilizzate per bypassare anche sistemi complessi come l'autenticazione a 2 fattori nel caso si utilizzi sms come secondo metodo di autenticazione.

6. Spesso i malintenzionati tentano l'opzione recupero password per trovare debolezze nel sistema. Instagram inizialmente permetteva di reimpostare la password con un semplice pin a 4 cifre inviato al telefono, con sole 9999 combinazioni possibili tale combinazione era facilmente bypassabile, resosi conto ovviamente tale procedura è stata disattivata ed il recupero password è più sicuro e macchinoso.

7.2 Soluzioni:

1. L'utente che desidera inserire una password manualmente, dovrà obbligatoriamente inserire una password di minimo 8 cifre contenente caratteri, caratteri speciali, numeri. Un algoritmo indicherà quanto la password sia forte, inoltre un messaggio di avviso informerà l'utente di non utilizzare password utilizzate altrove.
2. Verrà proibito l'utente di contenere nella propria password il nome utente e data di nascita.
3. Nel caso il nostro utente sia stato infettato da un keylogger grazie all'autenticazione a due fattori il malintenzionato dovrà disporre anche del

secondo metodo di accesso utilizzato per confermare la propria identità. La scelta di tale funzionalità sarà lasciata all'utente.

4. Ad ogni accesso con un nuovo dispositivo non registrato sarà inviata una notifica al proprio cellulare.
5. L' intercettazione dei messaggi ricevuti ed inviati sono implementati attraverso APIs, dunque e come fossero inviate dalla piattaforma principale su cui risiede anche l'annuncio, stessa cosa la ricezione. In parole povere l'implementazione di tale punto di sicurezza è già implementata da terzi. Le eventuali notifiche con sms, servono solo per reimpostare la password, dunque il malintenzionato dovrà obbligatoriamente usufruire della password della mail con cui si tenta il ripristino.
6. La password non verrà salvata in chiaro, ma hashata con algoritmo di hashing e criptata. All'utente sarà consentito di reimpostare la propria password attraverso l'email, per ricevere tale email bisognerà inserire un pin di 6 cifre contenenti caratteri misti.
7. FileSystem Criptato con la modalità spiegata al punto 6.

8.0 Boundary Conditions

Le boundary conditions per questo sistema potrebbero essere le seguenti:

1. Obbligo di registrazione per l'utente non registrato.
2. Limitazione delle operazioni per gli utenti senza partita IVA.
3. Funzionalità di login per accedere alla dashboard.
4. Visualizzazione degli annunci attivi in base alle piattaforme selezionate.
5. Possibilità di aggiungere, modificare e eliminare annunci dalla dashboard.
6. Funzione di ricerca per trovare rapidamente un annuncio.

7. Invio di notifiche periodiche e in tempo reale per eventi quali la vendita del prodotto.
8. Sezione profilo per visualizzare informazioni e scaricare la fattura o visualizzare il wallet.
9. Gestione delle transazioni tramite le API offerte dalle varie piattaforme.

Queste boundary condizioni descrivono i confini del sistema e definiscono le funzionalità e i limiti del sistema.

9.0 Glossario

@State: La variabile @state è una variabile utilizzata in swift per poter interagire con gli elementi di una view.

Api: (Application Programming Interfaces) sono interfacce di programmazione per le applicazioni che consentono a due sistemi software di comunicare tra loro. In pratica, le API forniscono un modo per accedere ai dati e alle funzionalità di un sistema da parte di altri sistemi o applicazioni. Questo permette agli sviluppatori di integrare funzionalità esterne nei loro prodotti, creare nuovi servizi e migliorare l'efficienza del lavoro.

Piattaforma: sistema software atto alla vendita online

Annunci: post inseriti dagli utenti che vogliono vendere i propri prodotti online

Fattura: documento commerciale che rappresenta una transazione tra due parti, in questo caso rilasciata agli utenti che effettuano vendite con partita iva

File system: utilizziamo un sistema di file e directory per memorizzare i dati critici del nostro sistema