

# SMARTPHONE STORE V 1.0

Relazione

*Progetto finale del corso  
“Programmazione ad Oggetti”*

Facoltà di “Ingegneria e scienze informatiche” sede  
di Cesena

Realizzato da:

**732952 – Silvia Colucci**

[silvia.colucci@studio.unibo.it](mailto:silvia.colucci@studio.unibo.it)

# **Smartphone Store**

## **Sistema per la gestione di un magazzino di smartphone**

*Silvia Colucci*

### **Analisi del problema**

L'applicazione è stata creata per gestire un magazzino di smartphone costituito da tre piani, ognuno dei quali contiene tre scaffali suddivisi in cinquanta ripiani. Attraverso quest'applicazione gestionale il personale potrà effettuare con estrema facilità le operazioni di aggiunta, vendita e ricerca di smartphones per poter soddisfare le richieste dei clienti.

In particolare tramite l'applicazione il personale è in grado di:

- Cercare uno smartphone specificandone il codice univoco.
- Cercare uno smartphone specificandone il nome, risalendo così alla sua collocazione.
- Inserire il nome di uno scaffale per visualizzare gli smartphone contenuti all'interno.
- Inserire uno smartphone in una determinata posizione del magazzino.
- Vendere uno smartphone.
- Avere un report sulla situazione complessiva del magazzino, con tutti gli smartphones presenti al suo interno nelle loro specifiche collocazioni.
- Avere un comando di logistica che permette al personale di accedere al piano desiderato per visualizzare gli scaffali contenenti gli smartphones.

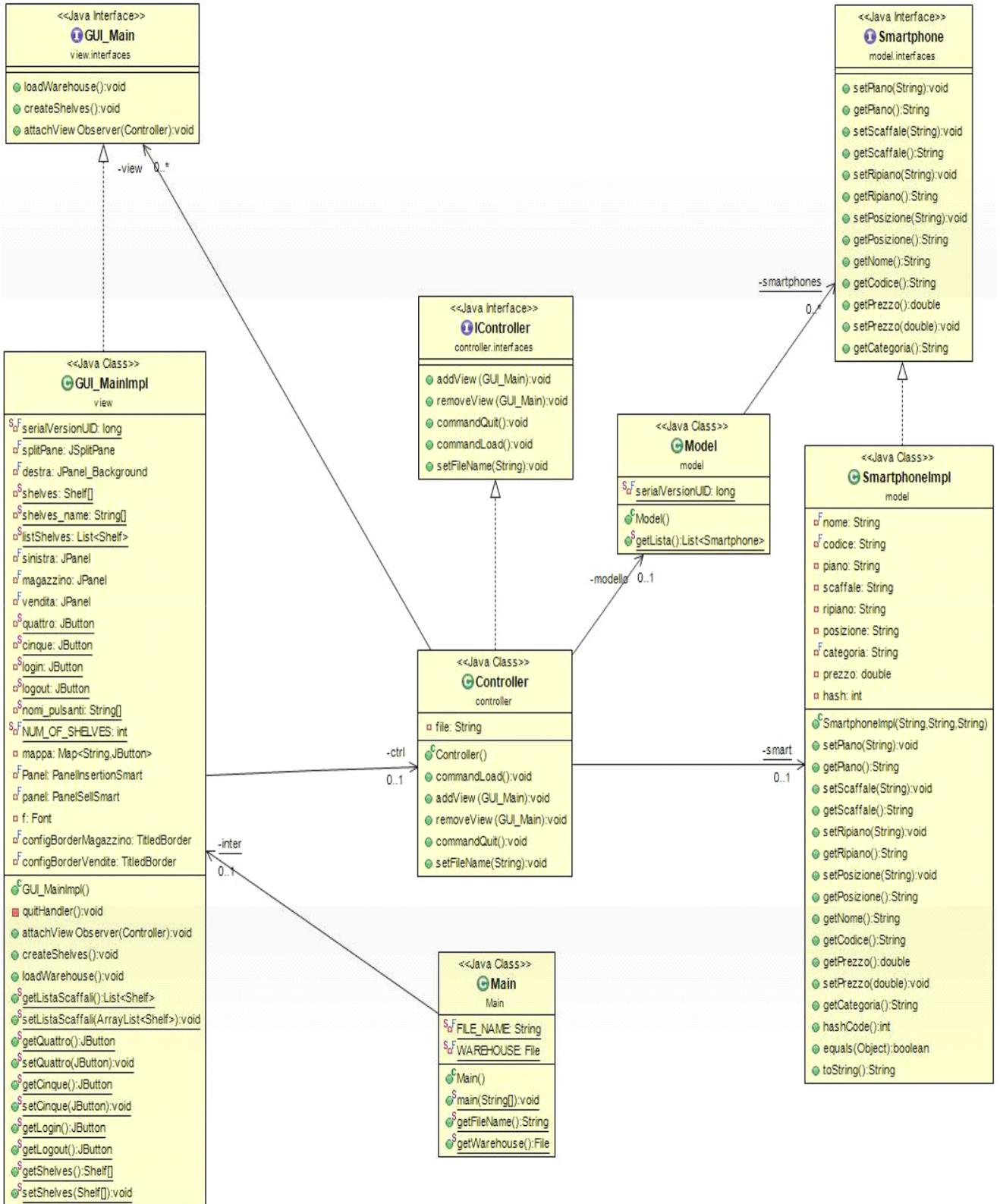
All'avvio del programma i pulsanti relativi al pannello di vendita sono disabilitati. Per abilitarli è necessario inserire un codice di protezione, cliccando su "login".

Alla pressione di "logout" questi verranno nuovamente disabilitati esattamente come all'avvio.

### **Progettazione architettonica**

L'applicazione è basata sul pattern Model-View-Controller. Questo permette di separare le parti che si occupano rispettivamente della conservazione dei dati, della visualizzazione degli stessi su schermo e della loro manipolazione, comandata tramite interfaccia grafica.

Segue l' UML introduttivo:



## **MODEL (Classi)**

Comprende la classe SmartphoneImpl, che crea l'oggetto smartphone costituito dal codice, dal modello, dalla categoria, dal prezzo insieme al piano, allo scaffale, al ripiano e alla posizione di appartenenza.

La classe PasswordGUIModel modella la GUI che abilita i pulsanti del pannello di vendita.

La classe Model estende la classe SmartphoneImpl e crea una lista statica di smartphone che sarà utilizzata da molte classi dell'applicazione.

## **VIEW(Classi)**

Comprende le seguenti classi:

### *-GUI\_MainImpl:*

Questa classe crea un pannello principale dividendolo in due sottopannelli di dimensioni fisse, "destra" e "sinistra"; Il pannello di sinistra è costituito da due sottomenù: uno per la gestione di ricerca all'interno del magazzino e l'altro per la gestione delle vendite.

Il pannello di destra è stato costruito come un JDesktopPane per poter inserire lo sfondo e per contenere le tabelle che simulano gli scaffali del magazzino.

### *-Shelf :*

E' la classe relativa alla creazione dello scaffale che è progettato come una tabella costituita da cinquanta righe e cinque colonne che simboleggiano i ripiani dell'interno.

### *-PanelInsertionSmart :*

La classe che implementa l'interfaccia Panel\_Insertion\_Smartphone, genera la GUI per l'inserimento di un nuovo smartphone che appare alla pressione del pulsante "New" del pannello vendita.

Questa comprende una serie di metodi privati, ognuno di essi effettua controlli su uno specifico campo della classe per garantire la correttezza dei dati in input.

Il metodo pubblico booleano check() consente all'utente di proseguire con l'aggiunta se tutti i controlli dei metodi privati sono verificati. Il metodo insertPosition() analizza i valori del pannello e verifica che la posizione dello scaffale in cui l'utente intende inserire il nuovo oggetto non sia

occupata, in tal caso attraverso il metodo `salvaSufileNewSmartphone()` il nuovo smartphone viene inserito nella tabella e scritto sul file.

*-PanelSellSmart :*

La classe crea la GUI per la vendita di uno smartphone. Dopo aver verificato che il codice inserito esiste nell'archivio e il prezzo corrisponde al rispettivo codice, lo smartphone viene venduto ed eliminato dal relativo scaffale nonché dal file.

*-TableResearchSmartphone :*

Da origine alla GUI relativa alla tabella generale che contiene tutte le informazioni relative agli smartphones ricercati, questa viene invocata dal tasto "Search from code" e dal pulsante "Search from shelf".

*-TableSmart'sName :*

E' la classe che genera la tabella contenente i nomi degli smartphones del magazzino; Viene caricata dopo la ricerca relativa al nome dello scaffale.

*-JPanel\_Background :*

Questa classe è utile per inserire gli sfondi del pannello principale, si ottiene estendendo `JDesktopPane` e implementando `paintComponent`.

*-SearchFromCode:*

Crea la finestra che appare alla pressione del pulsante "SearchFromCode", permettendo all'utente di inserire il codice dello smartphone da cercare: per ogni smartphone contenuto nella lista (`List<Smartphone>`), il codice inserito dall'utente viene confrontato con il corrispondente codice presente nel magazzino, se l'uguaglianza è verificata, l'applicazione fornisce una finestra mostrante la sua posizione.

*-SearchFromName:*

Da origine alla finestra che appare alla pressione del pulsante "SearchFromName", permettendo all'utente di inserire il nome dello smartphone da cercare. Allo stesso modo della classe precedente, se la lista di smartphone contiene il nome cercato, lo rende visibile all'interno della tabella dei nomi, con le quattro informazioni relative alla sua posizione.

*-SearchFromShelf:*

Crea la finestra che appare alla pressione del pulsante "SearchFromShelf", permettendo all'utente di inserire il nome dello scaffale del quale si vuole visualizzare il contenuto, se questo corrisponde a quello esistente apparirà una tabella contenente gli smartphones dello scaffale specificato.

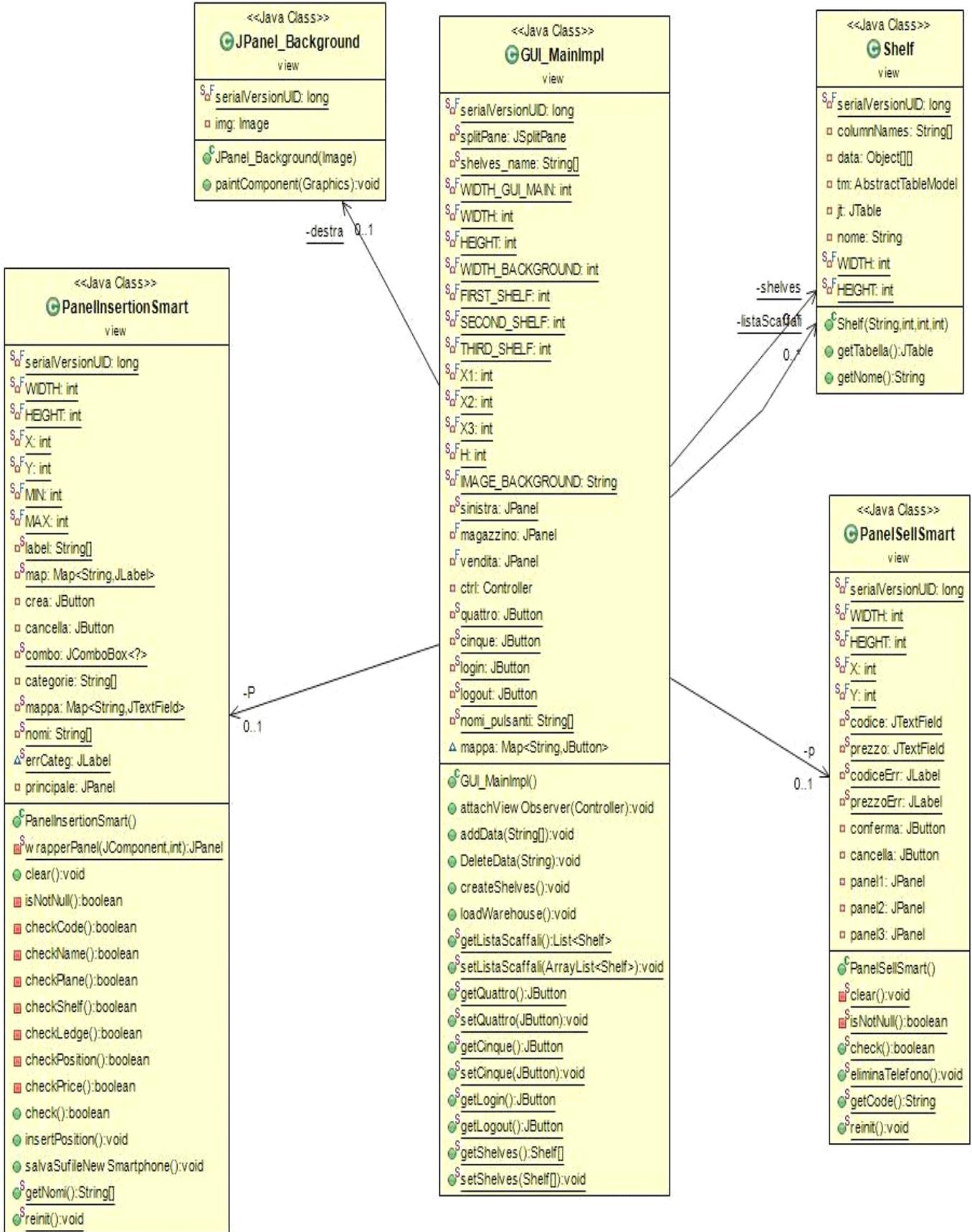
*-LogistGUI:*

Crea la finestra che appare alla pressione del pulsante "Logist", permettendo all'utente di inserire il piano del magazzino per visualizzare le tabelle contenenti gli smartphones disponibili nelle relative postazioni all'interno dei tre piani.

*-PasswordGUI:*

Crea la finestra che appare alla pressione del pulsante "login", con il fine di abilitare i pulsanti "NEW", "SELL" e "logout" del pannello vendita.

Sono state introdotte anche due classi: "Utility" e "Utilities" dove sono state dichiarate le costanti utilizzate prevalentemente nelle classi relative alla view per garantire una migliore leggibilità del codice e con il fine di evitare la dichiarazione sistematica delle medesime all'interno delle classi dell'applicazione che necessitano degli stessi valori.



## CONTROLLER (Classi)

Questo package comprende la classe ReportController che implementa ActionListener per il funzionamento del pulsante "Report" e la classe Controller. Nella suddetta classe dapprima viene creato il file di default "magazzino.dat", nel caso in cui questo non fosse già presente, e caricata la lista statica di smartphones attraverso il metodo commandLoad() del controller. Caricata la lista, l'interfaccia grafica principale sarà visibile. La classe Controller inoltre implementa i seguenti metodi dell'interfaccia IController:

- **void** addView(GUI\_Main v);

Per aggiungere la view a questo controller.

- **void** removeView(GUI\_Main v);

Per rimuovere la view da questo controller.

- **void** commandQuit();

Per uscire dall'applicazione quando l'utente intende chiuderla.

- **void** commandLoad();

Per consentire al programma di caricare la lista di smartphones presenti nel magazzino tramite il file legato all'applicazione.

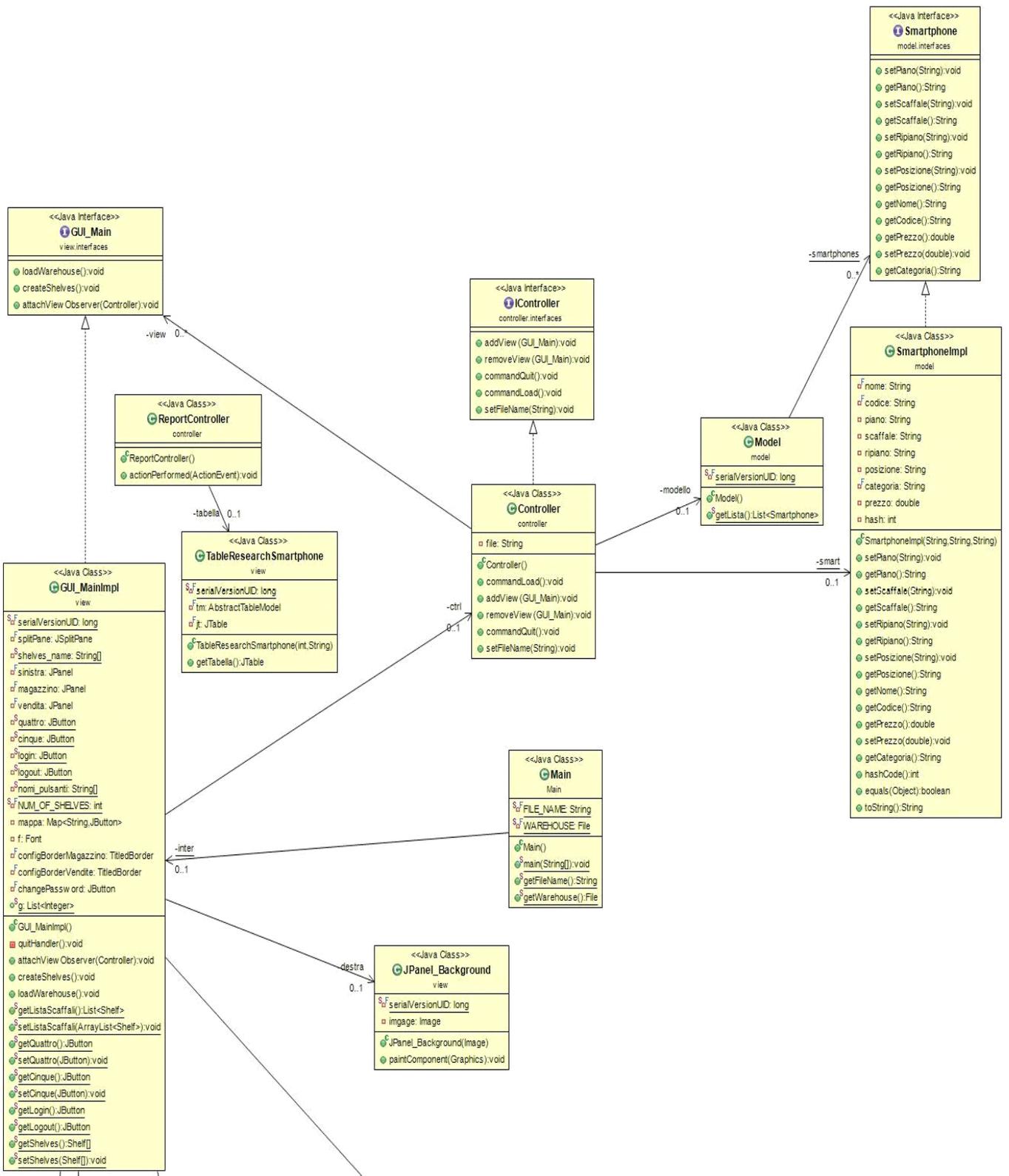
- **void** setFileName(String fileName);

Questo metodo serve alla classe Main per accedere al file dell'applicazione, caricare i dati e avviare il programma permettendo la visualizzazione dell'interfaccia grafica principale.

## MAIN

La classe Main avvia l'applicazione.

## **UML COMPLETO**





# Guida utente

Poche semplici regole per utilizzare SMARTPHONE\_STORE:

- All'avvio dell'applicazione è possibile cliccare su "Report" per visualizzare l'archivio del magazzino, cliccando sull'intestazione delle colonne la tabella sarà ordinata in base alle sue caratteristiche. Al primo avvio del programma il pulsante genererà una finestra vuota poichè non è ancora presente nessuno smartphone all'interno, il personale provvederà con le aggiunte.
- Per aggiungere o vendere smartphones bisogna effettuare il login, il codice di sicurezza è "12345".  
Qualora l'utente sbagliasse ad inserire i dati, sulla stessa finestra compariranno le giuste indicazioni per poterli inserire correttamente.
- Grazie al pulsante "Logist" si può accedere ai tre piani e visualizzare gli scaffali.
- Tramite i pulsanti "Search from name", "Search from code" e "Search from shelf" è possibile ricercare uno smartphone attraverso il nome, il codice o il nome dello scaffale e avere le informazioni che si desiderano. Il nome dello smartphone da cercare e quello dello scaffale da visualizzare possono essere scritti sia in maiuscolo che in minuscolo.

## NOTE FINALI

Considerando lo scopo del progetto precedentemente illustrato, il quale è funzionale alla facilitazione delle operazioni di gestione magazzino e di vendita da parte degli appositi operatori, l'idea iniziale è stata sviluppata strutturando un'architettura utile per le applicazioni gestionali. Tuttavia durante la realizzazione dello stesso, al fine di una maggiore completezza, è risultata vantaggiosa l'introduzione di tre ulteriori funzionalità riguardanti la logistica, il login/logout per poter accedere alle funzioni di aggiunta e di vendita, garantendo la sicurezza del sistema, ed infine il report riguardante l'archivio degli smartphones presenti in magazzino. Il flusso di lavoro è stato abbastanza lineare, e la durata complessiva riguardo la realizzazione dell'applicazione rientra nei termini indicati. Nonostante ci siano molti elementi da migliorare, lo svolgimento di tale progetto è risultato utile per una migliore comprensione del linguaggio e come spunto per futuri progetti.