



Object Design Document

Visual Assistant

Riferimento	VA_ODD
Versione	2.0
Data	04/02/2023
Destinatario	Top Management
Presentato da	Andrea Bucci Alessandro Cipullo Massimiliano Nunzio Gatta Michele Ginolfi Lorenzo Scorzelli Costantina Vincenzo
Approvato da	Rosario Di Palma Vincenzo Manserra

Revision History



Data	Versione	Descrizione	Autori
21/12/2022	1.0	Prima stesura	Andrea Bucci Alessandro Cipullo Massimiliano Nunzio Gatta Michele Ginolfi Lorenzo Scorzelli Costantina Vincenzo
04/02/2023	2.0	Revisione finale	Andrea Bucci



Sommario

Revision History.....	2
1. Introduzione.....	4
1.1 Linee Guida per la Scrittura del Codice.....	4
1.2 Definizione, acronimi e abbreviazioni.....	4
1.3 Riferimenti e Link utili.....	4
2. Packages.....	4
3. Class Interfaces.....	4
4. Class Diagram Ristrutturato.....	4
5. Elementi di riuso.....	4
5.1 Design Pattern usati.....	4
5.2 Componenti terzi.....	4
6. Glossario.....	4

1. Introduzione

1.1 Linee Guida per la Scrittura del Codice

Le linee guida includono una lista di regole che gli sviluppatori dovrebbero rispettare durante l'implementazione. Per l'implementazione di questo sistema, si è fatto riferimento alla convenzione di Effective Dart.

Dart: <https://dart.dev/guides/language/effective-dart/>

1.2 Definizioni, acronimi e abbreviazioni

Di seguito sono riportate alcune definizioni utili alla comprensione del documento:

- **Package:** Un meccanismo per organizzare classi in gruppi logici, principalmente (ma non solo) allo scopo di definire namespace distinti per diversi contesti;
- **Design Pattern:** Soluzione progettuale ad un problema, che permette il principio del riuso sia in fase di progettazione che in fase di sviluppo del software;
- **GUI:** Acronimo di Graphic-User-Interface che indica la rappresentazione visiva del software con la quale l'utente interagirà;
- **COTS:** Acronimo di Component-Off-The-Shelf che indica le componenti software e hardware che il mercato mette a disposizione per le aziende di sviluppo.

1.3 Riferimenti e Link utili

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura:

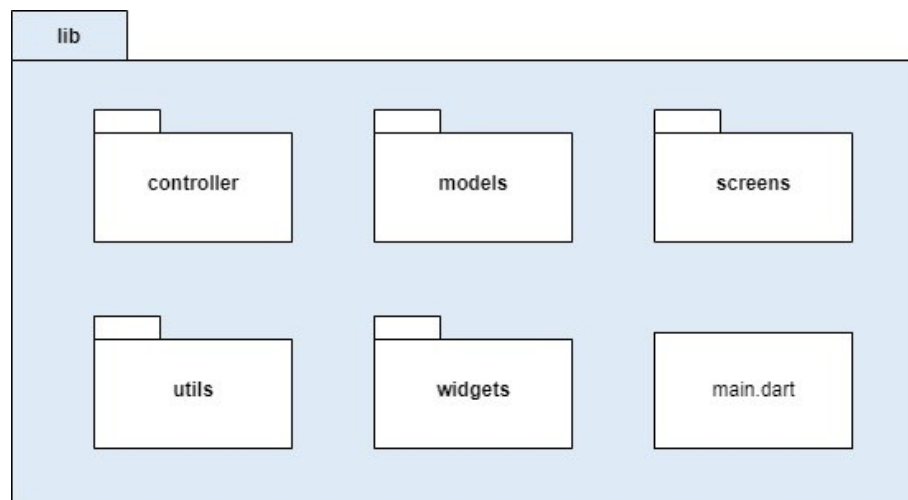
- Statement Of Work;
- Business Case;
- Requirement Analysis Document;
- System Design Document;
- Object Design Document;
- Test Plan;
- Matrice di tracciabilità;

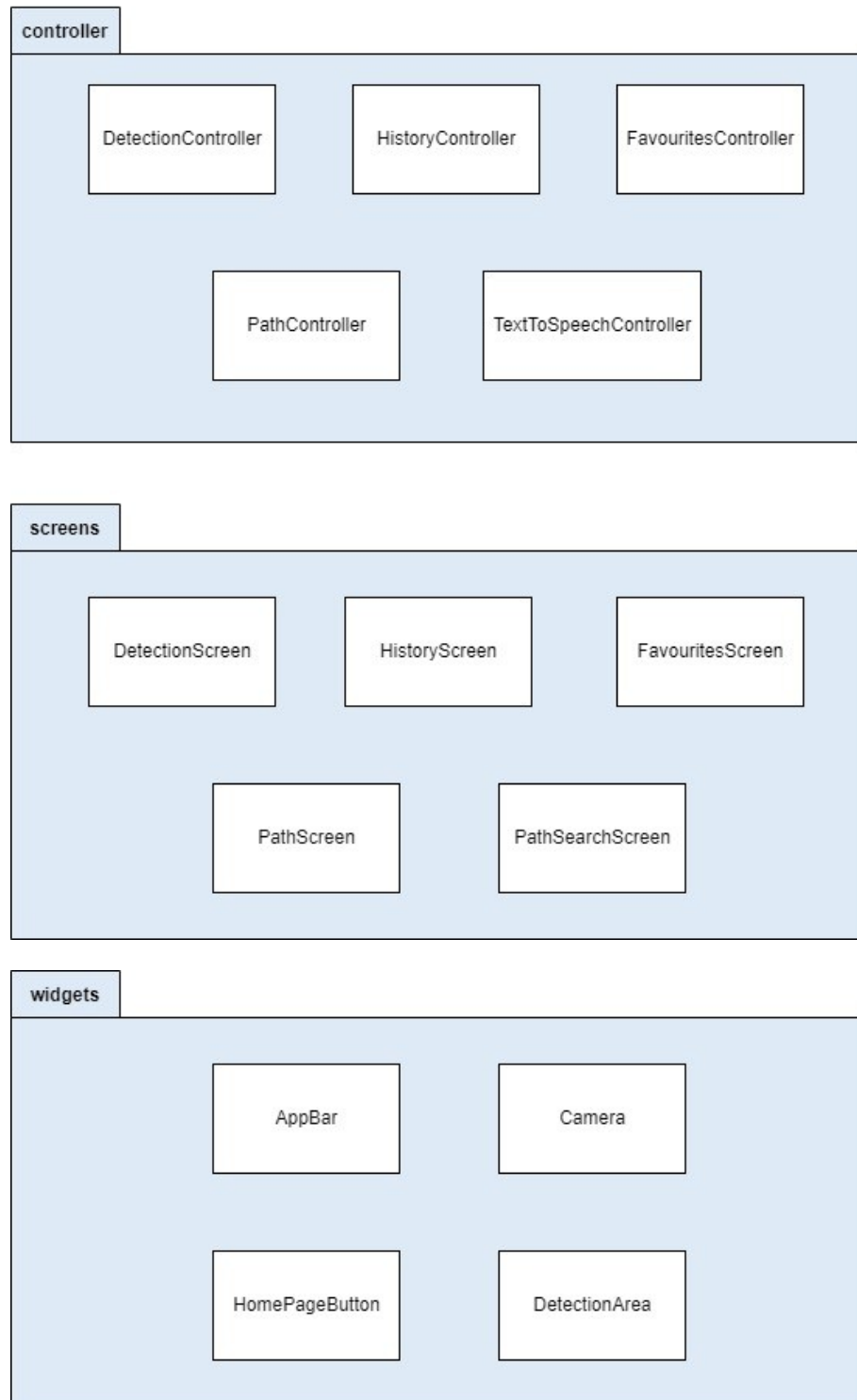
2. Packages

In questa sezione viene mostrata la suddivisione del sistema in package, in base a quanto definito nel documento di System Design. Tale suddivisione è motivata dalle scelte architetturelle prese e ricalca la struttura di directory standard definita da Dart.

La directory del progetto contiene:

- **assets** che contiene i modelli di riconoscimento e le immagini.
- **lib** è la cartella principale di implementazione e contiene:
 - **controller** che contiene tutte le interfacce del sistema.
 - **models** che contiene strutture dati e stringhe utili al sistema.
 - **screens** che contiene le interfacce grafiche dell'applicazione.
 - **utils** che contiene istanze di classi utili al sistema.
 - **widgets** che contiene frammenti riutilizzabili di interfaccia utente.
 - **main.dart** che rappresenta il punto di partenza del sistema.
- **test** che contiene i casi di test del sistema.
- **pubspec.yaml** che possiede i riferimenti alle dipendenze utilizzate e informazioni generali sul framework.

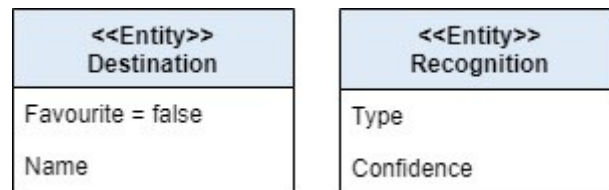




3. Class Interfaces

Nel nostro sistema non è stato necessario implementare delle interfacce, in quanto i metodi non vengono invocati da più di una classe, quindi sarebbe risultato ridondante e superfluo.

4. Class Diagram Ristrutturato



5. Elementi di riuso

5.1 Design Pattern usati

Nome: Singleton

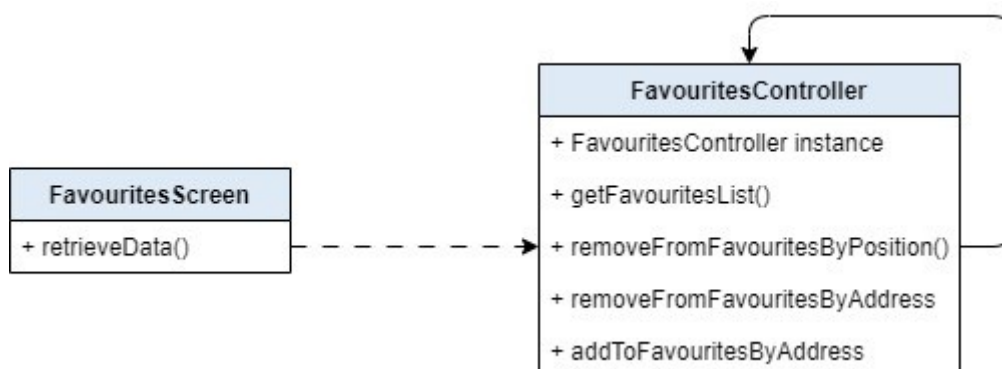
Descrizione: Il Singleton è un design pattern che ha lo scopo di garantire che venga creata una sola istanza di una determinata classe con un punto di accesso globale a tale istanza.

Soluzione: Il design pattern Singleton impedisce di accedere all'istanza di una determinata classe, poiché il costruttore è privato. Per poter accedere all'istanza è possibile invocare la proprietà **instance**, che è un riferimento all'istanza della classe stessa, oppure utilizzare il costruttore statico **getInstance()**.

Conseguenze:

- Accesso controllato all'unica istanza di una classe;
- Controllo completo di come e quando i client accedono all'interfaccia;
- Consente un raffinamento di operazioni e rappresentazioni;
- Mutua esclusione.

Esempio: Nel nostro sistema il design pattern Singleton è stato utilizzato insieme alla libreria **shared_preferences** per la gestione delle destinazioni preferite. In particolare, con il design pattern è possibile mantenere una sola istanza di controller dei preferiti, effettuando le operazioni di rimozione o aggiunta sulla stessa istanza.



5.2 Component Off-The-Shelf

Visual Assistant farà uso di Component-Off-The-Shelf, ovvero componenti software disponibili sul mercato per agevolare la realizzazione del progetto, con l'obiettivo di contenere i costi di sviluppo e manutenzione.

Il sistema utilizzerà Dart come linguaggio di programmazione a supporto della logica applicativa ed il framework Flutter per lo sviluppo delle GUI.

Per il riconoscimento degli elementi inquadrati dall'utente si utilizzerà *TensorFlow Lite*, una libreria open source per il machine learning.

Per la geolocalizzazione dell'utente e degli indirizzi si utilizzerà la libreria *Geolocator*, mentre per la ricerca dei percorsi *Google_Maps_Flutter*.

Per la gestione della cronologia e dei percorsi preferiti si è utilizzata la libreria *Shared Preferences*.

Infine, per l'input degli indirizzi da parte dell'utente si è utilizzato *Speech_To_Text* e per le note vocali fornite sistema Flutter *Text_To_Speech*.

6. Glossario

Singleton: Design pattern il cui scopo è quello di garantire che una determinata classe venga istanziata una sola volta, e quindi di fornire un punto di accesso globale a quell'istanza.

Class Diagram: Diagramma che rappresenta la struttura del sistema e viene usato per modellare i concetti del dominio del problema, interfacce e classi.

Percorso: Segmento di un itinerario che si adatta alle necessità di chi lo percorre.

Nota vocale: Comunicazione del sistema che utilizza l'impianto sonoro del dispositivo per aggiornare l'utente.