System Design Document

Visual Assistant

Riferimento	VA_SDD
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
30/11/2022	1.00	Prima stesura	Andrea Bucci Alessandro Cipullo Massimiliano Nunzio Gatta Michele Ginolfi Lorenzo Scorzelli Costantina Vincenzo
07/12/2022	1.01	Aggiunta dei Component Diagram e dei Servizi dei sottosistemi	Andrea Bucci Alessandro Cipullo Massimiliano Nunzio Gatta Michele Ginolfi Lorenzo Scorzelli Costantina Vincenzo
10/12/2022	1.1	Revisione	Massimiliano Nunzio Gatta Michele Ginolfi
04/02/2023	2.0	Revisione finale	Alessandro Cipullo

Sommario

1. Introduzione

1.1 Scopo del sistema

Si desidera realizzare un progetto che pone l'attenzione sulle persone con disabilità visive, consentendo loro di poter camminare in sicurezza. Lo scopo è quello di realizzare un sistema che guidi queste persone, aiutandole nella loro quotidianità ricorrendo ai servizi offerti da questa applicazione.

L'obiettivo principale di questo progetto, dunque, è quello di fornire un sistema che faciliti le persone non vedenti, elaborando un sistema che permetta di guidarle durante i movimenti della loro giornata. In particolare, il sistema proposto dovrà:

- Guidare vocalmente la persona disabile nei suoi spostamenti;
- Riconoscere gli ostacoli e avvisare la persona con un messaggio vocale;
- Avvisare vocalmente l'insorgere di pioggia o altre tipologie di allerta meteo;
- Riconoscere gli ostacoli e avvisare la persona con una vibrazione.

1.2 Design Goals

Rank	ID	Descrizione	Categoria	RNF di origine	Trade-offs
4	DG_01 - Response time	Il sistema dovrà garantire tempi di risposta rapidi, mediamente inferiori a 2 secondi, per rendere efficace l'interattività tra l'applicazione e l'utente.	Performance	RNF_PRE_01	Per garantire tempi di risposta rapidi, il sistema riconoscerà meno elementi contemporaneamente.
6	DG_02 - Throughput	Il sistema dovrà riuscire a riconoscere almeno 2 elementi allo stesso momento.	Performance	N/A	Per garantire il riconoscimento di più elementi contemporaneamente, i tempi di risposti saranno dilatati.
9	DG_03 - Memory	L'applicazione dovrà occupare al massimo 1 GB sulla memoria del dispositivo.	Performance	N/A	Lo spazio occupato dall'applicazione limiterà l'aggiunta di nuovi elementi riconoscibili.
7	DG_04 - Robustness	L'applicazione deve permettere all'utente di	Dependability	RNF_USA_01	N/A

		inserire nuovamente dati, qualora egli dovesse commettere degli errori.			
5	DG_05 - Availability	Il sistema dovrà risultare disponibile per l'utilizzo ogni qualvolta che si desidera, escludendo periodi di manutenzione e condizioni particolari come l'assenza di rete Internet.	Dependability	RNF_AFF_01, RNF_PRE_02	N/A
3	DG_06 - Fault Tolerance	L'applicazione deve essere in grado di effettuare un nuovo riconoscimento dell'elemento, qualora essa non dovesse riconoscere ciò che le viene inquadrato in fotocamera.	Dependability	RNF_AFF_02	Tentare più volte di riconoscere un elemento potrebbe aumentare i tempi di risposta, mettendo in pericolo l'utente.
1	DG_07 - Safety	Il sistema deve riconoscere le varie entità inquadrate dalla fotocamera con una precisione di riconoscimento almeno del 75% e con tempi di risposta entro i 3 secondi, in modo che, in presenza di un pericolo, l'utente abbia modo di reagire.	Dependability	RNF_AFF_02	N/A
11	DG_08 - Deployment Cost	L'applicazione non deve avere un costo per insegnare all'utente finale	Cost	RNF_PK_2	N/A

		come utilizzarla.			
12	DG_09 - Upgrade costs	L'applicazione non deve avere un costo di aggiornamento per eventuali cambiamenti di essa.	Cost	N/A	N/A
10	DG_10 - Estendibility	Deve poter essere semplice aggiungere nuovi elementi riconoscibili dall'applicazione.	Maintenance	N/A	L'aggiunta di nuovi elementi aumenterà lo spazio occupato dall'applicazione.
13	DG_11 - Portability	Il sistema dovrà essere sviluppato in maniera cross-platform, in modo tale da garantire un corretto funzionamento su piattaforme differenti (iOS e Android).	Portability	RNF_IMP_03	N/A
8	DG_12 - Utility	Il sistema deve permettere all'utente di usufruire della documentazione in formato audio.	End-User	RNF_USA_03	N/A
2	DG_13 - Usability	Il sistema deve permettere di interagire tramite comandi vocali alle sue funzionalità, e deve poter comunicare vocalmente con l'utente per guidarlo.	End-User	RNF_USA_01, RNF_USA_02	N/A

1.2.1 Trade-offs

Trade-off	Descrizione
Fault Tolerance vs Response Time	Tentare più volte di riconoscere un elemento potrebbe aumentare i tempi di risposta, mettendo in pericolo l'utente.
Response Time vs Throughput	Per garantire tempi di risposta rapidi, il sistema riconoscerà meno elementi contemporaneamente.
Memory vs Estendibility	Lo spazio occupato dall'applicazione limiterà l'aggiunta di nuovi elementi riconoscibili.

1.3 Acronimi

VA: Visual Assistant
DG: Design Goals
GUI: Graphic User Interface
CD: Class Diagram
CoD: Component Diagram
ER: Entity-Relationship
UC: Use Case

1.4 Riferimenti

- Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition
  Autori: Bernd Bruegge & Allen H. Dutoit
- Visual Assistant Requirement Analysis Document

1.5 Organizzazione del contenuto

Nella prima sezione, *Introduzione*, viene descritto in modo generale lo scopo del sistema e i design goals che il sistema propone di raggiungere.

La seconda sezione, *Architettura del sistema corrente*, descrive lo stato attuale dell'architettura (se presente).

La terza sezione, *Architettura del sistema proposto*, descrive il nuovo sistema attraverso la decomposizione in sottosistemi, il mapping hardware/software e la gestione dei dati persistenti.

Il *Glossario*, contiene la lista dei termini utilizzati nel documento.

2. Architettura del sistema corrente

Al momento, non esiste alcun software che implementi le funzionalità di Visual Assistant in un unico sistema. Il mercato non offre alternative complete al sistema da noi proposto, dunque, non esiste una reale architettura comparabile.

3. Architettura del sistema proposto

3.1 Panoramica

Il sistema proposto è basato sullo stile architetturale Two-Tier, un principio generale di architettura che può essere applicato a molti tipi differenti di sistema, a prescindere dalla tecnologia sottostante o dalla piattaforma. Nel caso di un sistema mobile, come il sistema da noi proposto, il livello di presentazione consiste nell'interfaccia utente e la logica di business (come previsto dal framework Flutter), e il livello di controllo che contiene le classi che recuperano i dati dalle API utilizzate nel sistema.

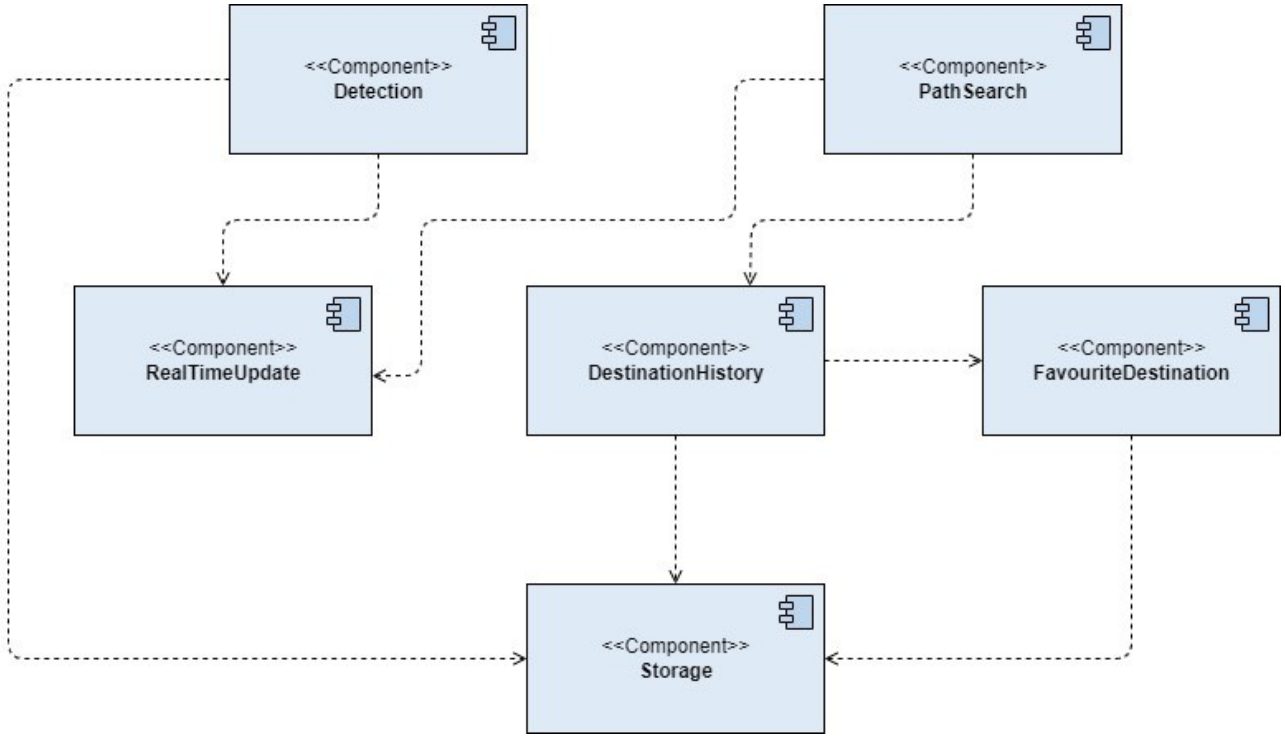
Nello sviluppo del sistema verrà usato il framework Flutter per la parte di front-end e la generazione delle viste. Per la logica applicativa, dunque il back-end, verrà usato Dart.

3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in sottosistemi

I sottosistemi individuati sono:

- **Riconoscimento**: Si occupa di gestire la funzionalità di riconoscimento di elementi come persone, animali, oggetti e pericoli.
- **RicercaPercorso**: Si occupa di ricercare un percorso in base alla destinazione scelta dall'utente.
- **AggiornamentoInTempoReale**: Si occupa delle funzionalità di aggiornamento in tempo reale delle indicazioni, delle condizioni meteo e della distanza di arrivo.
- **CronologiaDestinazioni**: Si occupa di gestire la funzionalità di mantenimento della cronologia delle ultime destinazioni immesse.
- **DestinazioniPreferite**: Si occupa di raggruppare in una lista le destinazioni preferite dell'utente.
- **Storage**: Si occupa di gestire il salvataggio della cronologia delle destinazioni e della lista dei preferiti.

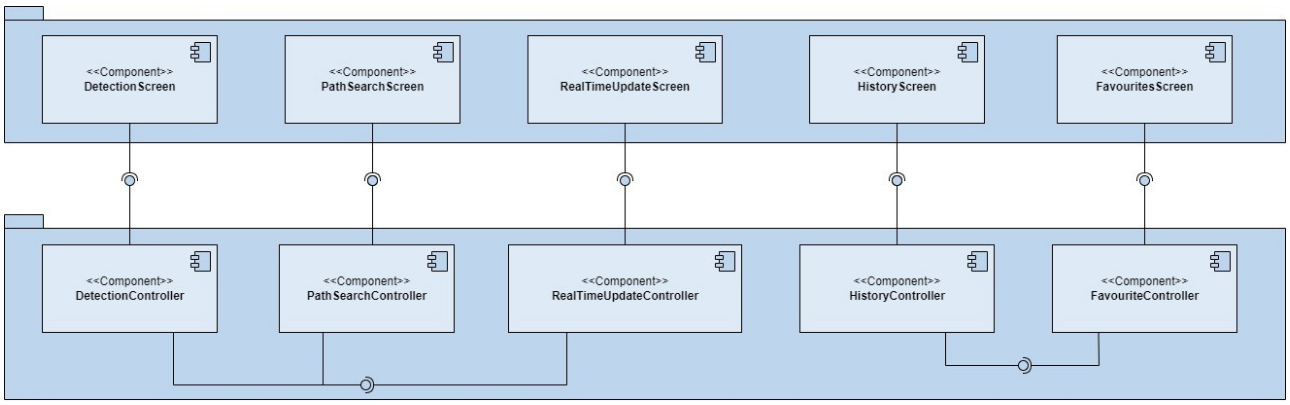


Di seguito, una vista dettagliata di ciascun sottosistema evidenziando le principali componenti:

- **Screen**: Contiene la vista mostrata al cliente.
- **Controller**: Si occupa della logica per il controllo del sistema.

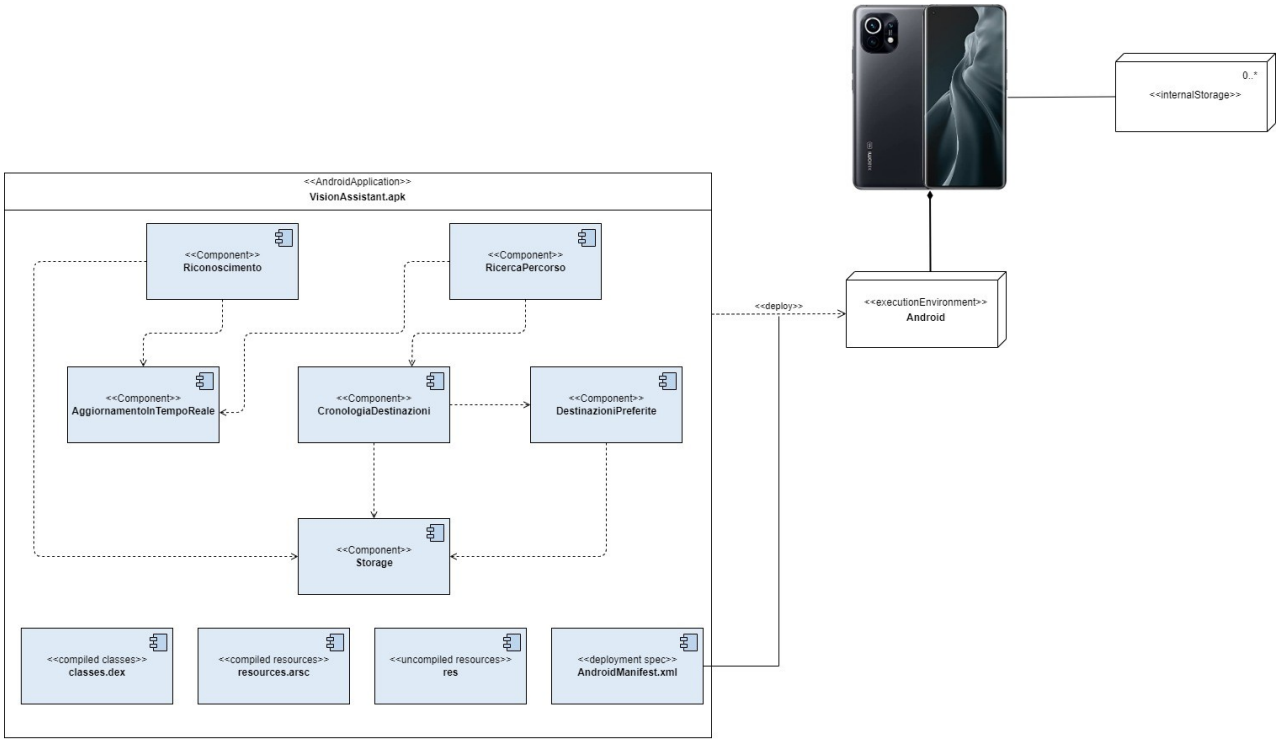
La scelta di utilizzare un'architettura a due tier, e non utilizzare un ulteriore tier per lo Storage, è giustificata dal fatto che i dati che il sistema salva in maniera persistente non sono numerosi. Pertanto, non risulta efficiente utilizzare un database di tipo relazionale, o eventualmente una nuova tecnologia come Google Firebase. Inoltre, tra le entità specificate nel Class Diagram non è presente alcuna relazione che giustifica l'utilizzo di una base di dati di tipo relazionale. Infine, Flutter, il framework utilizzato per implementare il sistema, possiede un'interfaccia di tipo `shared_preferences` che consente allo sviluppatore di salvare in modo persistente variabili o oggetti (attraverso una codifica in formato stringa e conseguente decodifica) da poter riutilizzare anche dopo la chiusura dell'applicazione.

3.2.2 Diagramma architetturale



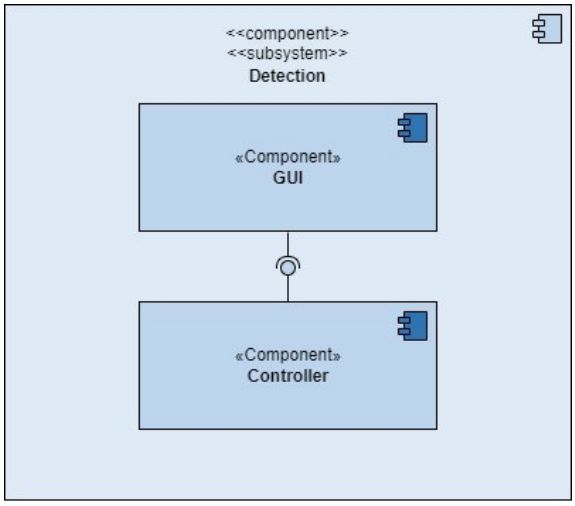
3.3 Mapping hardware/software

3.3.1 Deployment Diagram

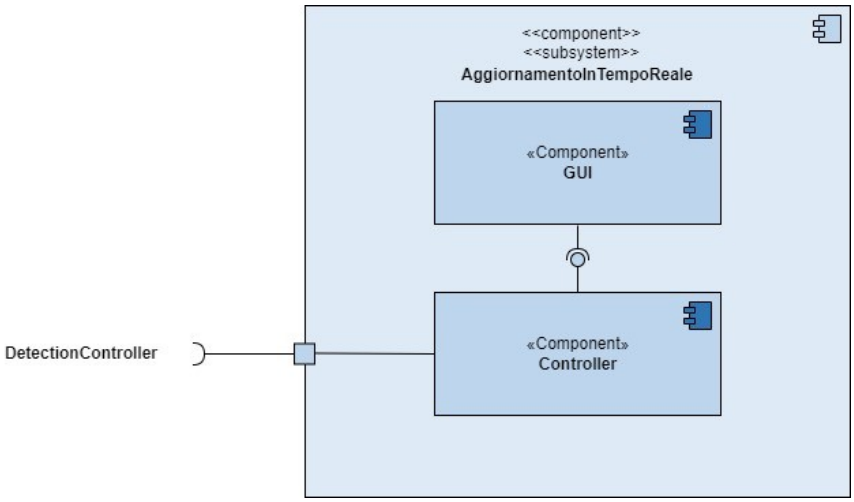


3.3.2 Component Diagram

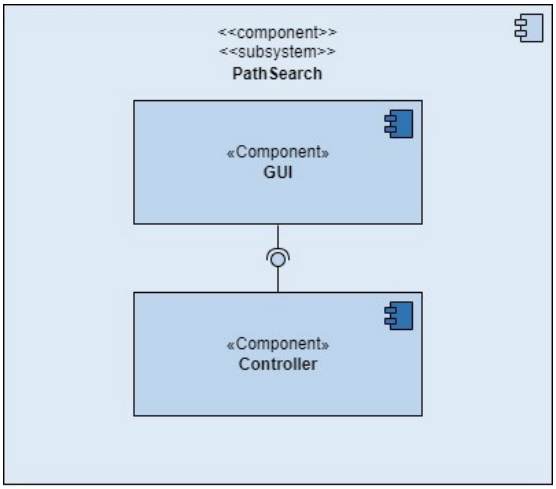
CoD – RICONOSCIMENTO



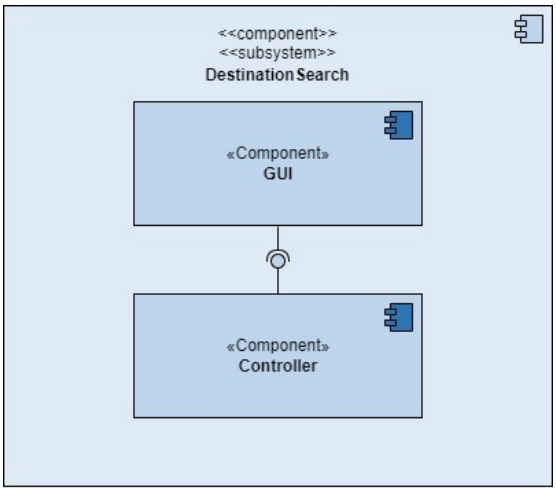
CoD – AGGIORNAMENTO IN TEMPO REALE



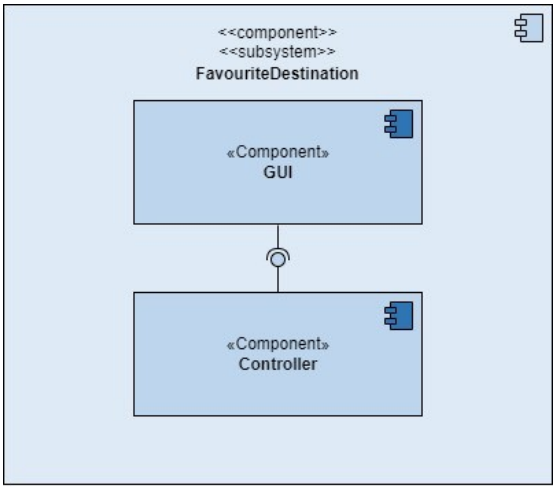
CoD – RICERCA PERCORSO



CoD – CRONOLOGIA DESTINAZIONI

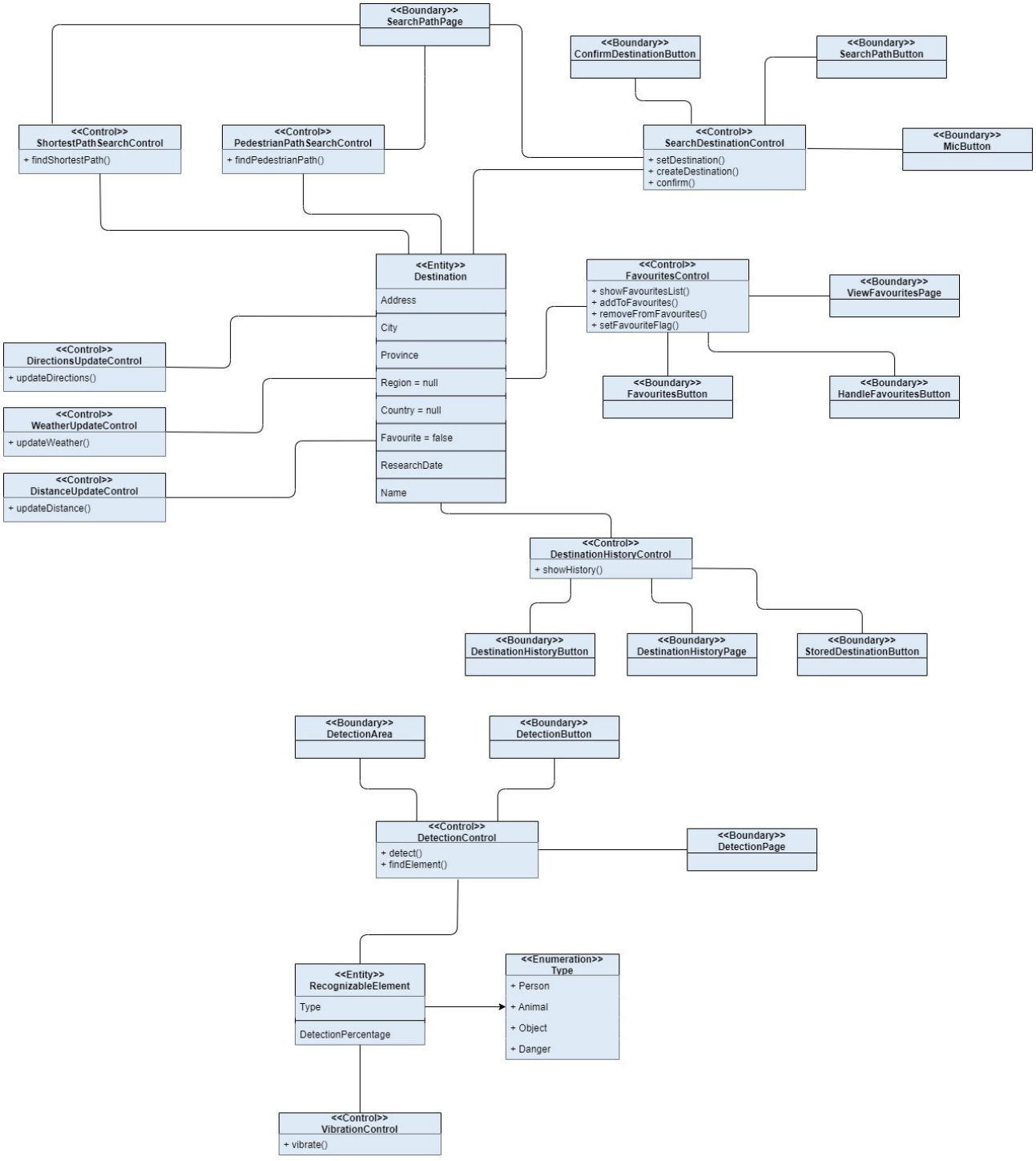


CoD – DESTINAZIONI PREFERITE



3.4 Gestione dati persistenti

3.4.1 Modello efficiente



3.5 Controlli accesso e sicurezza

Il sistema Visual Assistant è stato progettato per l'utilizzo da parte di persone con disabilità visive, e dunque, non capaci di interagire con sistemi software usuali. Pertanto, la scelta del progetto è stata quella di rendere il più facile possibile l'interazione tra l'utente e il sistema eliminando form e campi che potessero creare difficoltà all'utente.

Per questo motivo, non è presente un sistema di autenticazione e quindi l'utente può accedere a tutte le operazioni progettate per il sistema.

3.6 Controllo flusso globale del sistema

Nel sistema Visual Assistant, alcune funzionalità richiedono una interazione dell'utente attraverso l'interfaccia grafica. Per funzionalità specifiche come quella di riconoscimento, il sistema attende per eventi esterni ed agisce di conseguenza. Dunque, il sistema utilizzerà un meccanismo di controllo del flusso che è in parte "procedure-driven" e in parte "event-driven".

3.7 Condizioni limite

3.7.1 Start-up e configurazione

Identificativo **UCBC_01**	**Start-up e configurazione**	Data	07/12/2022
		Versione	1.00
		Autore	**Tutti i Team Member**
Descrizione	Lo UC permette l'avvio del sistema.		
Attore principale	**Utente**		
Attori secondari	NA		
Entry Condition	L'utente avvia l'applicazione.		
Exit Condition On success	Il sistema si avvia correttamente.		
Exit Condition On failure	Il sistema non si avvia.		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Utente:	Apre l'applicazione dal suo dispositivo.
2	Sistema:	Mostra all'utente la schermata principale.

3.7.2 Terminazione

Identificativo **UCBC_02**	**Terminazione**	Data	07/12/2022
		Versione	1.00
		Autore	**Tutti i Team Member**
Descrizione	Lo UC permette lo spegnimento del sistema.		
Attore principale	**Utente**		
Attori secondari	NA		
Entry Condition	L'utente avvia l'applicazione. AND L'applicazione è stata precedentemente avviata.		
Exit Condition On success	L'applicazione si chiude.		
Exit Condition On failure	L'applicazione non si chiude.		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Utente:	Esce dall'applicazione.
2	Utente:	Rimuove l'applicazione dal multi-tasking.

3.7.3 Fallimento

Identificativo **UCBC_03**	**Fallimento**	Data	07/12/2022
		Versione	1.00
		Autore	**Tutti i Team Member**
Descrizione	Lo UC definisce il comportamento del sistema in caso di fallimento.		
Attore principale	**Utente**		
Attori secondari	NA		
Entry Condition	L'utente si trova nella schermata di viaggio e non è connesso ad Internet.		
Exit Condition On success	NA		
Exit Condition On failure	Il sistema notifica l'assenza di connessione ad Internet ed invita l'utente a connettersi.		

FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Utente:	Si trova nella schermata di ricerca destinazione e cerca una destinazione.
2	Sistema:	Rileva l'assenza di connessione, avvisando l'utente.

4. Servizi dei sottosistemi

SOTTOSISTEMA RICONOSCIMENTO

Servizio	Descrizione
Riconoscimento persona	Questa funzionalità permette all'utente di riconoscere una persona.
Riconoscimento animale	Questa funzionalità permette all'utente di riconoscere un animale.
Riconoscimento oggetto	Questa funzionalità permette all'utente di riconoscere un oggetto.
Riconoscimento pericolo	Questa funzionalità permette all'utente di riconoscere un pericolo.

SOTTOSISTEMA RICERCA PERCORSO

Servizio	Descrizione
Ricerca del percorso più breve	Il servizio offre la possibilità di ricercare il percorso più breve considerando il punto di partenza e la destinazione inserita.
Ricerca del percorso con presenza di percorsi pedonali	Il servizio offre la possibilità di ricercare un percorso che include la presenza di un percorso pedonale considerando il punto di partenza e la destinazione inserita.

SOTTOSISTEMA AGGIORNAMENTO IN TEMPO REALE

Servizio	Descrizione
Aggiornamento in tempo reale delle indicazioni	Il servizio offre la possibilità di ricercare il percorso più breve considerando il punto di partenza e la destinazione inserita.
Aggiornamento in tempo reale delle condizioni meteo	Il servizio offre la possibilità di ricercare un percorso che include la presenza di un percorso pedonale considerando il punto di partenza e la destinazione inserita.
Aggiornamento in tempo reale della distanza di arrivo	Il servizio offre la possibilità di ricevere aggiornamenti in tempo reale sulla distanza che l'utente dovrà percorrere per arrivare a destinazione.

SOTTOSISTEMA DESTINAZIONI PREFERITE

Servizio	Descrizione
Visualizza destinazioni preferite	Questa funzionalità permette di poter visualizzare le destinazioni preferite.
Aggiungi ai preferiti	Questa funzionalità permette di aggiungere una destinazione ai preferiti.
Rimuovi dai preferiti	Questa funzionalità permette di eliminare una destinazione dai preferiti.
Rinomina destinazione	Questa funzionalità permette di dare un nome personalizzato alla destinazione.

SOTTOSISTEMA CRONOLOGIA DESTINAZIONI

Servizio	Descrizione
Visualizza cronologia destinazioni	Questa funzionalità permette di visualizzare la cronologia delle destinazioni.
Aggiungi ai preferiti	Questa funzionalità permette di aggiungere una destinazione della cronologia ai preferiti.
Rimuovi dai preferiti	Questa funzionalità permette di eliminare una destinazione dai preferiti.

5. Glossario

Visual Assistant: Nome dell'applicazione che si andrà a realizzare.

Component Diagram: Diagramma UML che ha lo scopo di rappresentare la struttura interna del sistema software modellato in termini dei suoi componenti principali e delle relazioni fra essi.

Design Goal: Descrive una funzionalità che gli sviluppatori vorrebbero ottimizzare.

SDD: Acronimo di System Design Document. È un documento che tratta nel dettaglio della progettazione del sistema e dei suoi obiettivi.

Architettura: Organizzazione di base di un sistema, espressa dai suoi componenti, dalle relazioni tra di loro e con l'ambiente, e i principi che ne guidano il progetto e l'evoluzione.

Deployment Diagram: Diagramma UML che mostra l'architettura di esecuzione di un sistema, inclusi i nodi come gli ambienti di esecuzione hardware o software, e il middleware che li collega.

Framework: Sistema che consente di estendere le funzionalità del linguaggio di programmazione su cui è basato, fornendo allo sviluppatore una struttura coerente ed efficace al fine di effettuare azioni e comandi in modo semplice e veloce.

Trade-off: Compromesso fra due o più funzionalità o fra l'utente ed il sistema, dove l'ottimizzazione di uno va a discapito dell'altro.