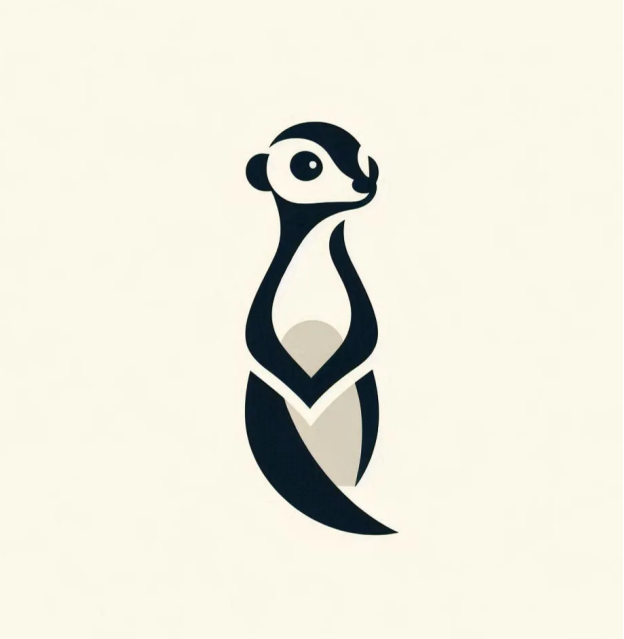
Object Design Document (ODD)

MeerKat

Object Design Document

Versione 0.1



Coordinatore del progetto:

Nome	Matricola

Revision History

Data	Versione	Descrizione	Autore
28/11/2024	0.1	Scrittura indice.	Stefano Nicolò , Francesco Giuseppe Trotta, Giuseppe

Data	Versione	Descrizione	Autore
			Ballacchino, Gabriel Tabasco
2/12/2024	0.1.1	Introduzione	Gabriel Tabasco
10/12/2024	0.1.2	Sezione Pacchetti	Giuseppe Ballacchino,Gabriel Tabasco
11/12/2024	0.1.3	Interfacce delle classi	Francesco Giuseppe Trotta, Stefano Nicolò Zito

Indice

1. Introduzione

1.1 Trade-off nel design degli oggetti

- Descrizione delle decisioni principali prese nel design:
  - Memoria vs Tempo di risposta: Si punta a metodi di accesso che prioritizzano velocità di risposta rispetto alla quantità di memoria utilizzata.
  - Costruire vs Comprare: *Utilizzo di strutture dati e framework esistenti dove possibile.*

1.2 Linee guida per la documentazione delle interfacce

- Convenzioni:
  - **Nomi delle classi:** Singolare (es. `User`, `ChatRoom`).
  - **Nomi dei metodi:** Frasi verbali (es. `getNome`, `inviaMessaggio`).
  - **Eccezioni:** Restituite tramite meccanismi dedicati.

1.3 Definizioni e abbreviazioni

- **UUID:** Identificatore univoco universale.
- **Byte array:** Rappresentazione binaria di dati (es. immagini).
- **DateTime:** Tipo di dato per rappresentare date e orari.

1.4 Riferimenti

- Specifiche dei requisiti (RAD).
- Diagrammi UML correlati.

---

2. Pacchetti

2.1 Descrizione generale

- **Pacchetto** `UserManagement`**:** Gestisce utenti, membri e project manager.
- **Pacchetto** `Chat`**:** Gestisce chatroom e messaggi.
- **Pacchetto** `TeamManagement`**:** Gestisce squadre e attività.

2.2 Dipendenze tra pacchetti

- `UserManagement` è utilizzato da `Chat` e `TeamManagement` per la gestione degli utenti.
- `Chat` dipende da `UserManagement` per associare messaggi agli utenti.

2.3 Organizzazione dei file

- Struttura dei file:

---

3. Interfacce delle classi

3.1 Class: `User`

- **Attributi:**
  - `+nome: String`
  - `+cognome: String`
  - `+email: String`
  - `+password: String`
  - `+dataNascita: Date`
  - `+id: UUID`
  - `+immagine: byte[]`
- **Metodi:**
  - Getter:
    - `+getNome(): String`
    - `+getCognome(): String`
    - `+getEmail(): String`
    - `+getPassword(): String`
    - `+getDataNascita(): Date`
    - `+getId(): UUID`
    - `+getImmagine(): byte[]`
  - Setter:

        - `+setNome(nome: String): void`
        - `+setCognome(cognome: String): void`
        - `+setEmail(email: String): void`
        - `+setPassword(password: String): void`
        - `+setDataNascita(dataNascita: Date): void`
        - `+setId(id: UUID): void`
        - `+setImmagine(immagine: byte[]): void`

3.2 Class: `UserViewModel`

- **Metodi:**
    - `+CreaUtente(user: User): void`
    - `+AggiornaUtente(user: User): void`
    - `+EliminaUtente(userId: UUID): void`
    - `+GetUtente(userId: UUID): User`

3.3 Class: `UserView`

- **Metodi:**
    - `+VisualizzaUtente(user: User): void`
    - `+VisualizzaListaUtenti(utenti: List<User>): void`

3.4 Class: `ProjectManager`

- **Eredita da:** `User`
- **Metodi aggiuntivi:**
    - `+assegnaAttivita(): void`

3.5 Class: `ProjectManagerViewModel`

- **Metodi:**
    - `+CreaProjectManager(projectManager: ProjectManager): void`
    - `+AggiornaProjectManager(projectManager: ProjectManager): void`
    - `+EliminaProjectManager(projectManagerId: UUID): void`

- `+GetProjectManager(projectManagerId: UUID): ProjectManager`

3.6 Class: `ProjectManagerView`

- **Metodi:**
  - `+VisualizzaProjectManager(projectManager: ProjectManager): void`
  - `+VisualizzaListaProjectManager(projectManagers: List<ProjectManager>): void`

3.7 Class: `Member`

- **Eredita da:** `User`
- **Attributi aggiuntivi:**
  - `+listaAttivita: List<User>`
- **Metodi aggiuntivi:**
  - `+completaAttivita(): bool`

3.8 Class: `MemberViewModel`

- **Metodi:**
  - `+CreaMembro(member: Member): void`
  - `+AggiornaMembro(member: Member): void`
  - `+EliminaMembro(memberId: UUID): void`
  - `+GetMembro(memberId: UUID): Member`

3.9 Class: `MemberView`

- **Metodi:**
  - `+VisualizzaMembro(member: Member): void`
  - `+VisualizzaListaMembri(membri: List<Member>): void`

3.10 Class: `ChatRoom`

- **Attributi:**
  - `+membri: List<User>`
  - `+messaggi: List<Message>`
- **Metodi:**

- `+inviaMessaggio(): void`

3.11 Class: `ChatRoomViewModel`

- **Metodi:**
  - `+CreaChatRoom(chatRoom: ChatRoom): void`
  - `+AggiornaChatRoom(chatRoom: ChatRoom): void`
  - `+EliminaChatRoom(chatRoomId: UUID): void`
  - `+GetChatRoom(chatRoomId: UUID): ChatRoom`

3.12 Class: `ChatRoomView`

- **Metodi:**
  - `+VisualizzaChatRoom(chatRoom: ChatRoom): void`
  - `+VisualizzaListaChatRoom(chatRooms: List<ChatRoom>): void`

3.13 Class: `Message`

- **Attributi:**
  - `+mittente: User`
  - `+timestampMessaggio: DateTime`
  - `+testoMessaggio: String`
  - `+statoMessaggio: String`

3.14 Class: `MessageViewModel`

- **Metodi:**
  - `+CreaMessaggio(message: Message): void`
  - `+AggiornaMessaggio(message: Message): void`
  - `+EliminaMessaggio(messageId: UUID): void`
  - `+GetMessaggio(messageId: UUID): Message`

3.15 Class: `MessageView`

- **Metodi:**
  - `+VisualizzaMessaggio(message: Message): void`

- `+VisualizzaListaMessaggi(messages: List<Message>): void`

3.16 Class: `Team`

- **Attributi:**
  - `+nomeSquadra: String`
  - `+membri: List<User>`

---

4. Conclusione

- Questo documento deve essere mantenuto aggiornato durante lo sviluppo.
- Usare come riferimento durante la fase di test e integrazione.