VisualCue

# Indice

[1 Indice 2](#_Toc188991029)

[2 Introduzione 3](#_Toc188991030)

[2.1 Informazioni sul progetto 3](#_Toc188991031)

[2.2 Abstract 3](#_Toc188991032)

[2.3 Scopo 3](#_Toc188991033)

[3 Analisi 4](#_Toc188991034)

[3.1 Analisi del dominio 4](#_Toc188991035)

[3.2 Analisi e specifica dei requisiti 4](#_Toc188991036)

[3.3 Use case 5](#_Toc188991037)

[3.4 Pianificazione 6](#_Toc188991038)

[3.4.1 Analisi 6](#_Toc188991039)

[3.4.2 GUI 6](#_Toc188991040)

[3.4.3 Configurazioni ambiente di sviluppo 6](#_Toc188991041)

[3.4.4 Algoritmo 6](#_Toc188991042)

[3.5 Analisi dei mezzi 8](#_Toc188991043)

[3.5.1 Software 8](#_Toc188991044)

[3.5.2 Hardware 8](#_Toc188991045)

[3.5.3 Web server 8](#_Toc188991046)

[4 Progettazione 9](#_Toc188991047)

[4.1 Design procedurale 9](#_Toc188991048)

[4.2 Design dei dati e database 9](#_Toc188991049)

[4.3 Design delle interfacce 10](#_Toc188991050)

[5 Implementazione 11](#_Toc188991051)

[6 Test 12](#_Toc188991052)

[6.1 Protocollo di test 12](#_Toc188991053)

[6.2 Mancanze/limitazioni conosciute 12](#_Toc188991054)

[7 Consuntivo 12](#_Toc188991055)

[8 Conclusioni 14](#_Toc188991056)

[8.1 Sviluppi futuri 14](#_Toc188991057)

[8.2 Considerazioni personali 14](#_Toc188991058)

[9 Bibliografia 15](#_Toc188991059)

[9.1 Sitografia 15](#_Toc188991060)

[10 Glossario 15](#_Toc188991061)

[11 Indice delle figure 16](#_Toc188991062)

[12 Allegati 16](#_Toc188991063)

# Introduzione

## Informazioni sul progetto

Allievo: Riccardo Deiuri

Docente: Geo Petrini

Luogo: Scuola Arti e Mestieri di Trevano, TI

Sezione: Informatica

Data di inizio: 04.02.2025

Data di consegna/fine: 04.04.2025

## Abstract

The creation of the VisualCue web application revolutionizes the way users memorize and learn new terms by combining text and images into customizable flashcards. Users can create flashcards with custom content by uploading their own or having them generated. Entire collections of cards can also be automatically generated from user-supplied text input, with images created by a built-in algorithm.

VisualCue offers several test modes, such as memory matching, multiple choice, and timed challenges, to accommodate different learning styles. Users can organize their papers into private or public collections and engage in real-time challenges, fostering a competitive and interactive learning environment.

This innovative application leverages visual association to enhance memory retention, making it an effective and engaging tool for students of all ages and subjects.

## Scopo

Il principale obiettivo di questo progetto è fornire una metodologia di studio efficace e intuitiva, perfetta per chi desidera migliorare l'uso della memoria attraverso le immagini. Al centro dell'applicazione VisualCue ci sono le card, strumenti fondamentali che consentono di studiare e memorizzare un termine scritto associandolo a un'immagine. Questa tecnica si dimostra estremamente utile in diverse materie scolastiche e trova applicazione anche nel mondo del lavoro, ad esempio per apprendere termini tecnici, acronimi e altro ancora. A rendere il tutto ancora più coinvolgente è l'introduzione di un sistema di punti, che aggiunge un elemento di divertimento e stimola maggiormente l'utente nello studio.

L'obiettivo secondario di questo progetto è acquisire esperienza nella gestione e nello sviluppo di un'applicazione di questa portata. Questo lavoro mi darà l'opportunità di approfondire l'uso di Python e del framework Flask, che finora ho utilizzato solo marginalmente. Inoltre, avrò la possibilità di esplorare per la prima volta React, un framework molto utilizzato in ambito professionale.

# Analisi

## Analisi del dominio

L’applicazione web in sviluppo è una piattaforma dedicata allo studio, accessibile a utenti di tutte le età. Sebbene esistano già soluzioni simili ben strutturate, VisualCue si distingue per un'interfaccia grafica fortemente incentrata sull'impatto visivo, favorendo la memorizzazione attraverso le immagini.

Gli utenti potranno accedere al sito tramite e-mail da qualsiasi browser. Tuttavia, la connessione ottimale è garantita principalmente in Europa, dove si trova il server.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **Req-001** | |
| **Nome** | Creazione interfaccia |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di creare l’interfaccia grafica dell’applicativo web |
| **Sotto requisiti** | |
| **Req-001\_1** | Creare pagina “Home” |
| **Req-001\_2** | Creare pagina “Create” |
| **Req-001\_3** | Creare pagina “Test Memory” |
| **Req-001\_4** | Creare pagina “Test Combine” |
| **Req-001\_5** | Creare pagina “Profile” |
| **Req-001 6** | Creare pagina “Login/Sing-up” |

|  |  |
| --- | --- |
| **Req-002** | |
| **Nome** | Creare e configurare webserver |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di realizzare il webserver con tutto il necessario (vedere sotto requisiti) |
| **Sotto requisiti** | |
| **Req-002\_1** | Creare utenti sul server |
| **Req-002\_2** | Installare e configurare MySQL |
| **Req-002\_3** | Configurare IP e SSH |
| **Req-002\_4** | Installare Node |
| **Req-002\_5** | Installare Python |

|  |  |
| --- | --- |
| **Req-003** | |
| **Nome** | Creazione database |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di creare il database |
| **Sotto requisiti** | |
| **Req-003\_1** | Creare utenti |
| **Req-003\_2** | Creare tabelle |

|  |  |
| --- | --- |
| **Req-004** | |
| **Nome** | Algoritmo per “Home” |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di creare l’algoritmo per far funzionare la sezione Home, ovvero la visualizzazione delle collezioni pubblicate da altri utenti. |

|  |  |
| --- | --- |
| **Req-004** | |
| **Nome** | Algoritmo per “Create” |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di creare l’algoritmo per far funzionare la sezione “Create”, ovvero la creazione e/o la generazione di una collezione di card. |
| **Sotto requisiti** | |
| **Req-004\_1** | Implementare la possibilità di aggiungere un file di testo come input |
| **Req-004\_2** | Implementare la possibilità di aggiungere il testo direttamente in un input textarea. |
| **Req-004\_3** | Implementare la generazione delle immagini. |
| **Req-004\_4** | Implementare la possibilità di aggiungere da parte dell’utente delle immagini. |
| **Req-004\_5** | Implementare la possibilità di poter modificare le collezioni e le card. |

|  |  |
| --- | --- |
| **Req-005** | |
| **Nome** | Eliminazione delle collezioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Gli utenti possono eliminare le proprie collezioni. |

|  |  |
| --- | --- |
| **Req-006** | |
| **Nome** | Modifica delle collezioni |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Gli utenti possono modificare le proprie collezioni. |
| **Sotto requisiti** | |
| **Req-006\_1** | Si necessita di eliminare, aggiungere, modificare le card della collezione. |

|  |  |
| --- | --- |
| **Req-007** | |
| **Nome** | Implementazione login e signup |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di implementare il login e signup |

|  |  |
| --- | --- |
| **Req-008** | |
| **Nome** | Criptazione dati sensibili |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di criptare i dati sensibili, come ad esempio la password dell’utente |

|  |  |
| --- | --- |
| **Req-009** | |
| **Nome** | Connessione database |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di connettere il database in modo da utilizzarlo per l’applicativo web. |

|  |  |
| --- | --- |
| **Req-010** | |
| **Nome** | Modifica del profilo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Gli utenti possono modificare le proprie informazioni personali. |

|  |  |
| --- | --- |
| **Req-011** | |
| **Nome** | Algoritmo per “Test” |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di creare l’algoritmo per far funzionare la sezione Test, ovvero l’esecuzione dei test con la collezione selezionata. |
| **Sotto requisiti** | |
| **Req-011\_1** | Implementare il test “Memory” |
| **Req-011\_2** | Implementare il test “Combine multiple-choises” |
| **Req-011\_3** | Implementare il test “Combine” |
| **Req-011\_4** | Implementare il test “Combine on time” |
| **Req-011\_5** | Implementare il conteggio del punteggio. |

|  |  |
| --- | --- |
| **Req-012** | |
| **Nome** | Configurazione sicurezza del server |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di configurare il server in modo che sia il più sicuro possibile |
| **Sotto requisiti** | |
| **Req-012\_1** | Monitoraggio delle richieste e stato dei servizi del server. |

|  |  |
| --- | --- |
| **Req-013** | |
| **Nome** | Algoritmo per “Profile” |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessita di creare l’algoritmo per far funzionare la sezione “Profile”, ovvero la visualizzazione delle informazioni dell’utente. |
| **Sotto requisiti** | |
| **Req-013\_1** | Implementare la possibilità di modificare le proprie informazioni personali |
| **Req-013\_2** | Implementare la possibilità di visualizzare le proprie collezioni. |

|  |  |
| --- | --- |
| **Req-014** | |
| **Nome** | Sfida real-time |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Gli utenti possono sfidare altri utenti per una sfida a tempo in tempo reale. |

|  |  |
| --- | --- |
| **Req-015** | |
| **Nome** | Stampare collezioni |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Note** | Si necessita di poter stampare le collezioni. |

**Spiegazione elementi tabella dei requisiti:**

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case

I casi d’uso rappresentano l’interazione tra i vari attori e le funzionalità del prodotto.

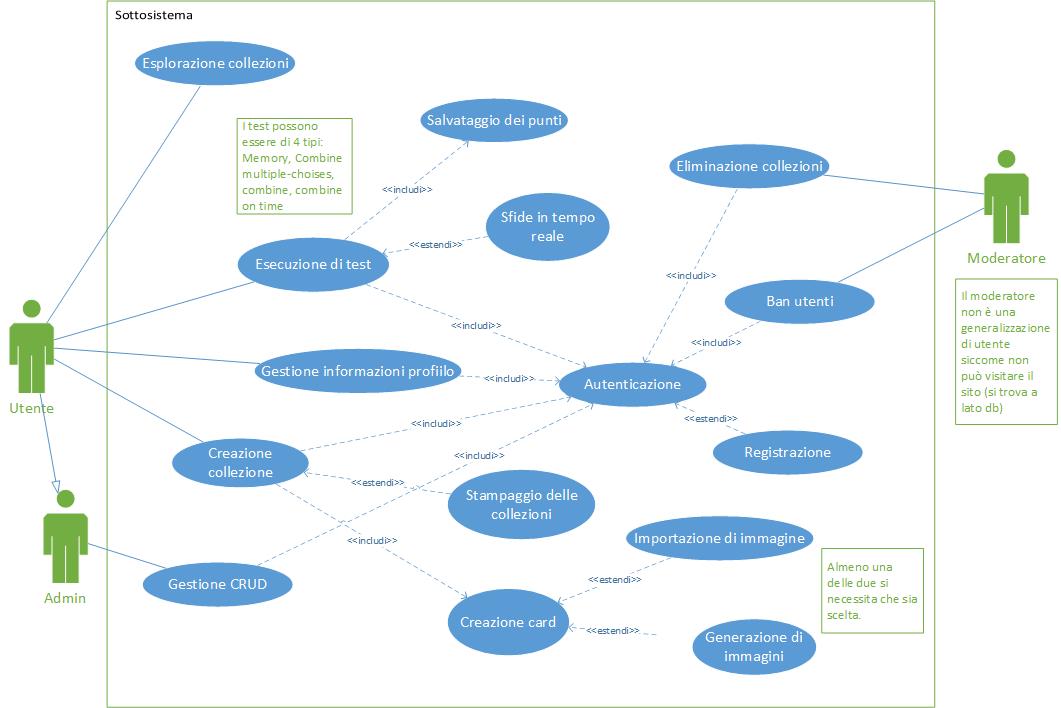


Figura 1 Use case

## Pianificazione

La pianificazione di questo progetto è stata fatta tramite il Gantt preventivo utilizzando il metodo di pianificazione agile, la pianificazione tramite Gantt viene usata come se fosse una linea guida essendo che le tempistiche vengono segnalate dalle milestone (sprint) definiti insieme al committente del progetto.

Il progetto suddiviso in cinque parti fondamentali che sono:

* Analisi
* GUI
* Configurazioni varie
* Algoritmo
* TestCase

Tuttavia, oltre ai testcase e il margine finale in ogni attività è calcolato un piccolo margine e un momento di test e di documentazione.

### Analisi

La parte di analisi serve per capire le specifiche che deve avere il progetto discusse con il committente, creare l’Abstract e creare gli schemi del progetto che sono:

* Qdc
* Gantt
* Use case
* Diagramma ER
* Activity diagram

### GUI

La parte di GUI (interfaccia grafica) serve per creare le interfacce del nostro applicativo, inizialmente le pagine vengono solo disegnate (parte design) con mockflow per poi essere implementate tramite il framework React.

### Configurazioni ambiente di sviluppo

La parte di configurazioni serve per installare il server e configuralo in modo da essere utilizzato, lo stesso discorso vale per nodeJS e python che verrà installato e configurato sul server.

### Algoritmo

La parte di algoritmo serve per definire la parte back-end del codice, configurare nodeJS (linguaggio utilizzato), implementare le varie funzionalità che avrà il nostro applicativo, far comunicare le pagine fra di loro e testare tutto ciò che è stato implementato.

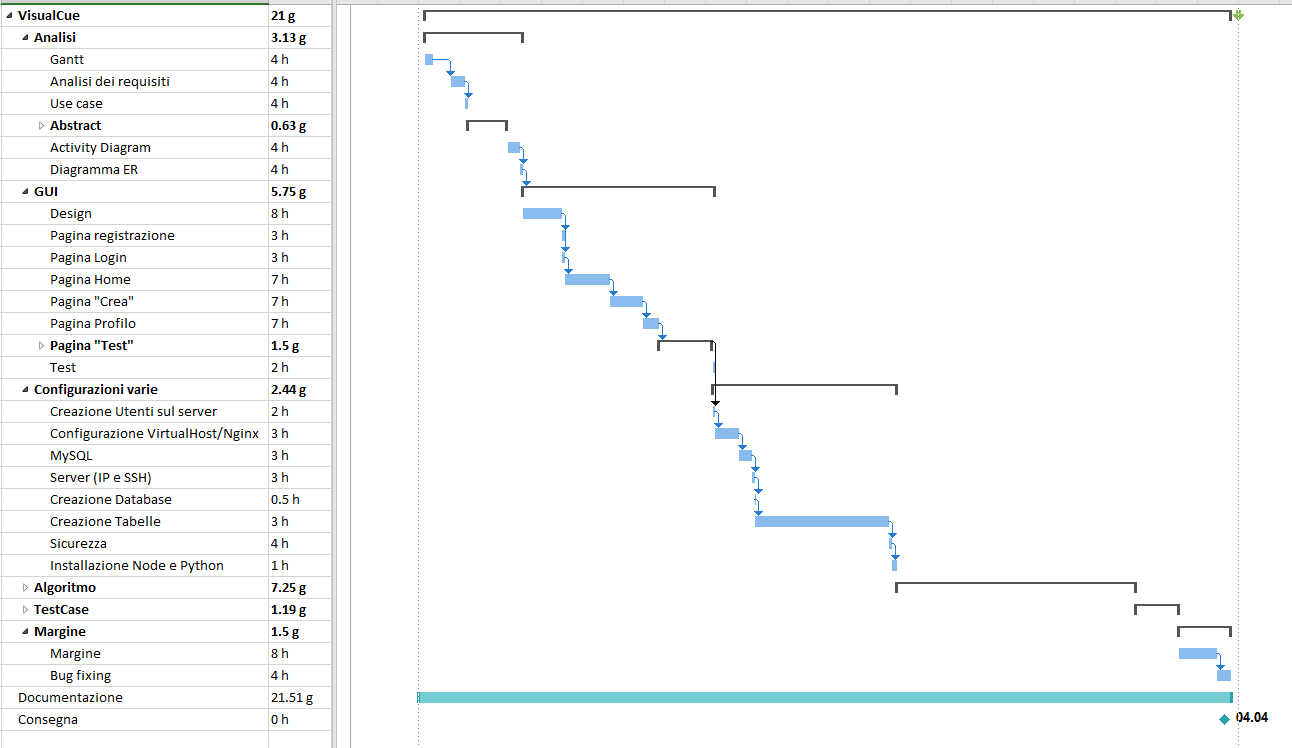


Figura 2 Gantt preventivo

## Analisi dei mezzi

### Software

Visual Studio Code: versione 1.87.2

Browser Google Chrome (indifferente il browser)

Front-end

* React: versione
* Tailwind CSS: versione

Back-end:

* Python: versione
* Flask: versione
* SQLAlchemy: versione

Database:

* MySQL: versione

### Hardware

#### Hardware Sviluppo

Gli Hardware di sviluppo sono dei computer forniti dalla scuola con queste specifiche:

* OS: Windows 10, 64 bit
* Processore: Intel i7-7700, 3.60GHz, 4 Cores
* Ram: 16 GB, 2400 MHz
* Scheda grafica: Intel HD Graphics 630, 8 GB
* Display: HP E24i G4, 1920x1200, 60Hz

#### Hardware Prodotto

Come Hardware prodotto serve una macchina che si possa collegare a internet e un browser per poter accedere al sito del nostro applicativo

### Web server

* Node js e Python
* Memoria: 146 GiB
* Processore: Intel(R) Xeon(R) CPU E5640 @ 2.67GHz

# Progettazione

## Design procedurale

A seguire sono rappresentati gli activity diagram dei principali algoritmi dell’applicativo (Test Memory, Create).

*Test Memory*

Una volta nella pagina dei test, l’utente deve selezionare la collezione con cui desidera effettuare il test. Successivamente, sceglie il tipo di test, in questo caso “memory”. A questo punto, il programma raccoglie tutte le card della collezione, separando ciascuna in due parti: una con il testo e l’altra con l’immagine. Le card vengono poi mescolate e mostrate all’utente. L’utente può selezionare una coppia di card. Se abbina correttamente il testo all’immagine corrispondente, la coppia viene segnalata come corretta. In caso di errore, può riprovare fino a trovare l’abbinamento giusto per ogni coppia.

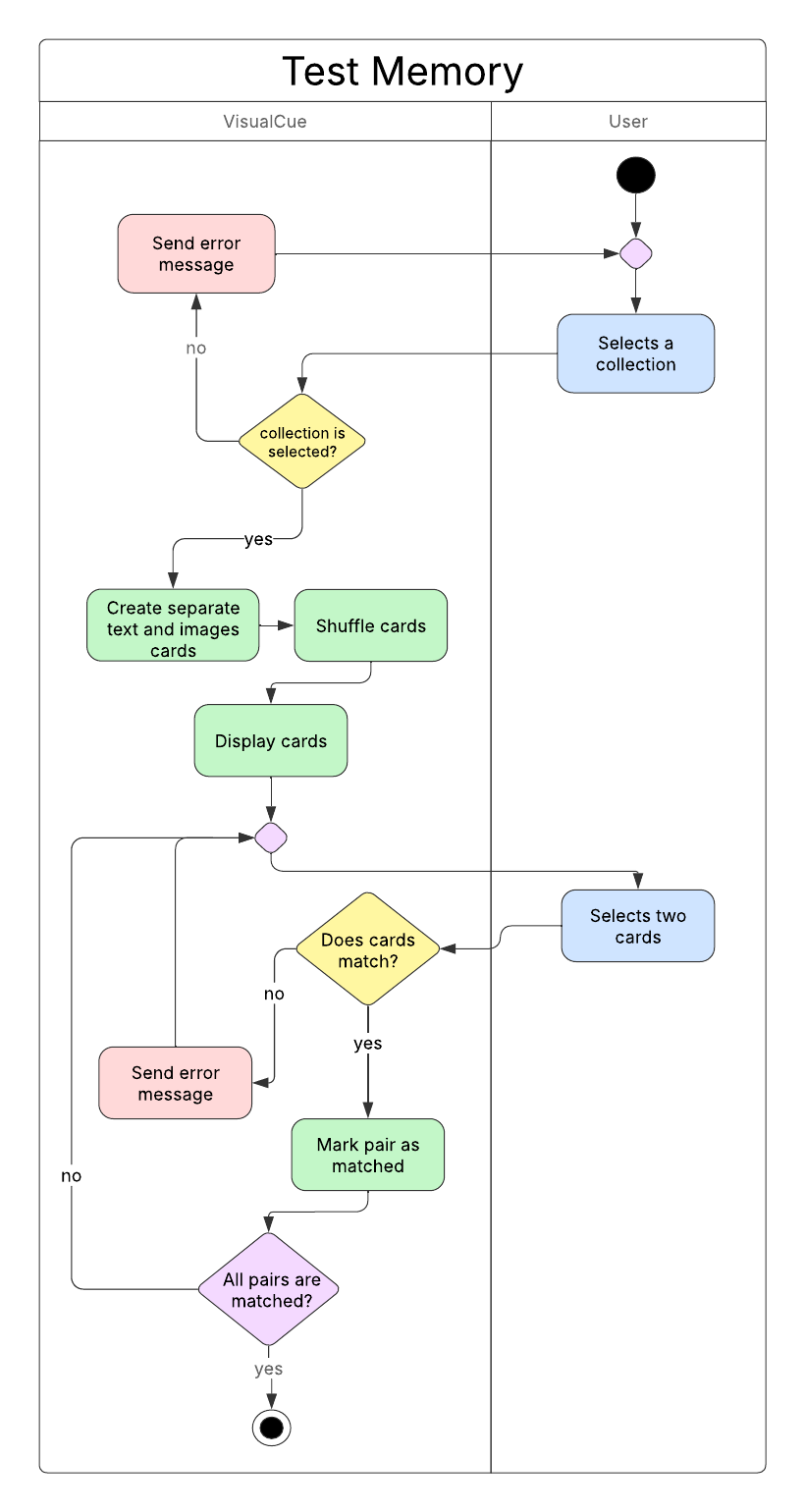


Figura 3 Activity Diagram - Test Memory

*Create (Manualmente)*

Il diagramma illustra il processo di creazione manuale di una collezione e delle relative card.

L’utente inizia fornendo i dati per la collezione. Il sistema verifica la validità di queste informazioni e, se corrette, procede con la creazione della collezione. In caso di errore, l’utente viene invitato a correggere i dati. Successivamente, l’utente può inserire i dati per una card. Anche in questo caso, il sistema controlla la correttezza delle informazioni prima di procedere con la creazione della card. Se i dati non risultano validi, appare un messaggio di errore e l’utente deve apportare le necessarie correzioni. Dopo aver creato la card, l’utente ha la possibilità di aggiungerne un’altra selezionando “Next card” (in questo caso la pagina si aggiorna per consentire l’inserimento di una nuova card) oppure di concludere il processo selezionando “Finish”.

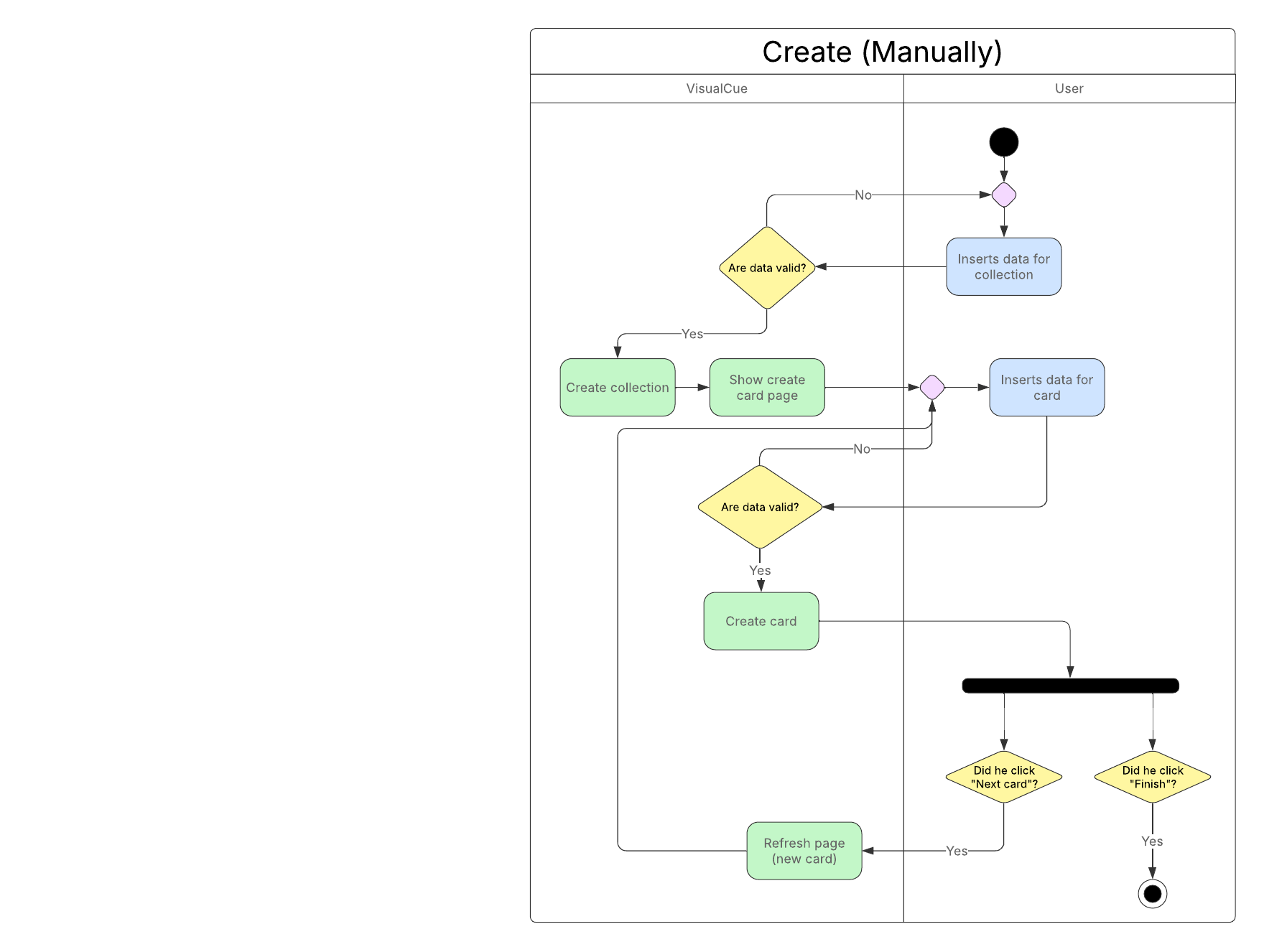


Figura 4 Activity Diagram - Create

## Design dei dati e database

Descrizione delle entità:

* **user**: memorizza le informazioni degli utenti, identificati in modo univoco tramite l'email.
* **collection**: rappresenta un insieme di card create dagli utenti. Ogni collezione è associata a un utente e può essere pubblica o privata.
* **card**: ogni card appartiene a una collezione e include un testo e un'immagine. All'interno della stessa collezione, le card non possono avere nomi duplicati (vincolo di unicità).
* **test**: rappresenta un test effettuato da un utente su una collezione. Contiene il punteggio ottenuto e il tipo di test.
* **type\_test**: definisce i vari tipi di test disponibili (es. Memory, Combine).

Relazioni principali:

* Un utente può creare più collezioni.
* Una collezione contiene più card.
* Un utente può eseguire test su diverse collezioni.

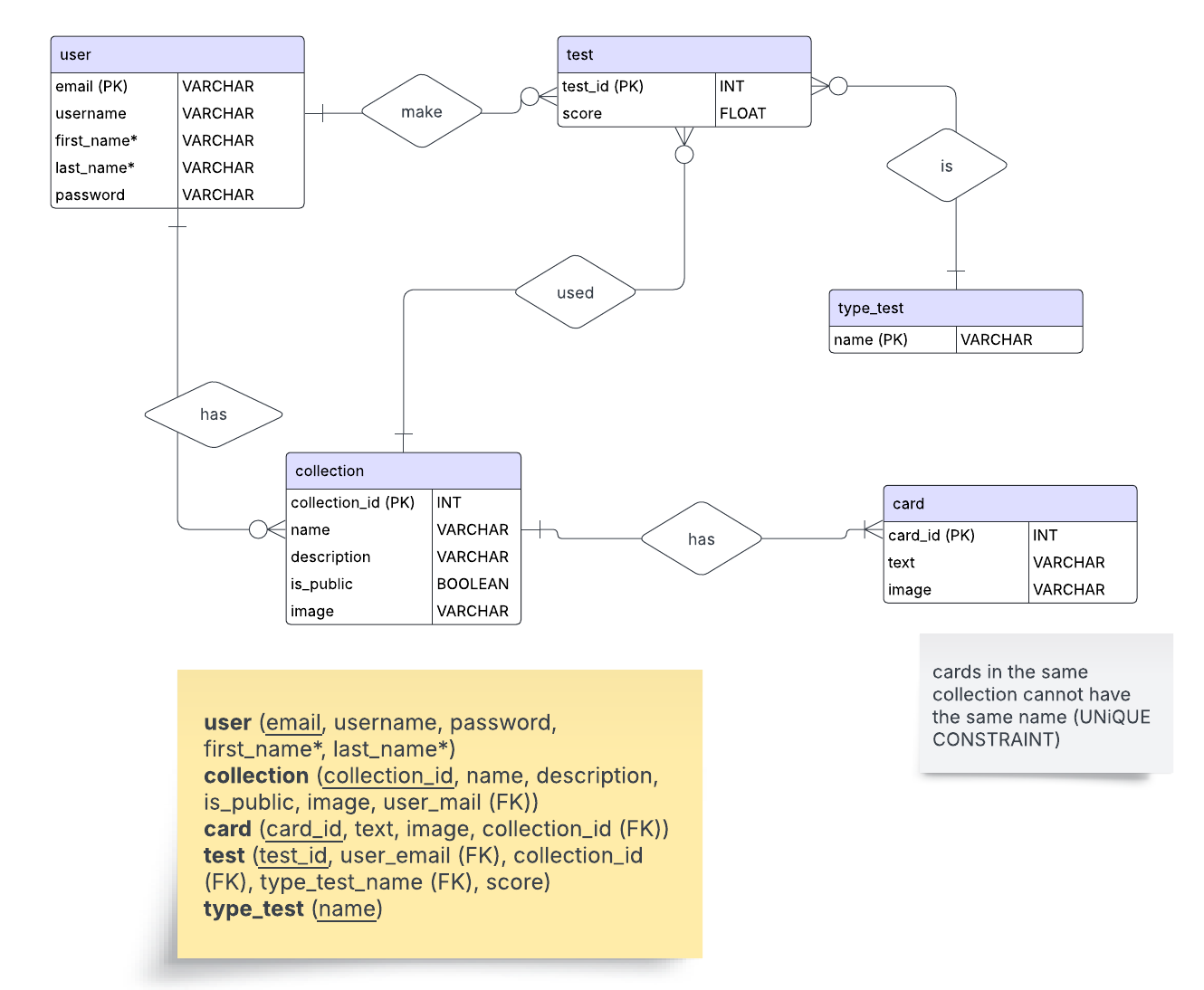


Figura 5 diagramma Er

## Design delle interfacce

# Implementazione

# Test

## Protocollo di test

## Mancanze/limitazioni conosciute

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 3.4)

Figura 6 Gantt consuntivo

# Conclusioni

## Sviluppi futuri

## Considerazioni personali

# Bibliografia

## Sitografia

* <https://mockflow.com/>.
* <https://getbootstrap.com/>
* <https://www.w3schools.com/>
* <https://stackoverflow.com/>
* <https://www.npmjs.com/>
* <https://chat.openai.com/>
* <https://www.youtube.com/?hl=it&gl=IT>
* <https://it.wikipedia.org/wiki/>
* <https://lucid.app/>

# Glossario

|  |  |
| --- | --- |
| **Termine** | **Significato** |
|  |  |
|  |  |
| VirtualHost | Programma utilizzato per la creazione e l’uso di macchine virtuali |
| Nginx | Web server leggero e ad alte prestazioni, che può essere usato anche come reverse proxy, load balancer, cache HTTP e proxy di posta elettronica |
|  |  |
|  |  |
| NAS | dispositivo di archiviazione dedicato ai file che rende i dati sempre disponibili ai dipendenti per collaborare in maniera efficace in una rete |
| Javascript | è un linguaggio di programmazione che gli sviluppatori utilizzano per realizzare pagine Web interattive |
| NodeJs | È un runtime system open source multipiattaforma orientato agli eventi per l'esecuzione di codice JavaScript |

# Indice delle figure

# Allegati

* Diari di lavoro
* Codici sorgenti
* Macchine virtuali
* Qdc
* Abstract
* Database
* Activity Diagram
* Gantt
* Gantt consuntivo
* Use Case
* Diagramma ER
* Foto logo
* Foto design interfaccie
* Prodotto