

Università degli Studi di Messina

Progetto di ***:INGEGNERIA DEL SOFTWARE***

VR FOR ALL



SOTWARE DEVELOPMENT DOCUMENT

Realizzato da :

CERENZIA FRANCESCO & MADDOCCO ALESSANDRO

Sommario

1. Descrizione progetto...

2. Panoramica software

2.1 Linguaggi per la descrizione UML.

2.2 Tecnologie implementative

2.4 Team di sviluppo

2.5 Requisiti Team

2.6 Tempistiche di progetto.

2.7 Requisiti Informali

3. MODELLO DI SVILUPPO & CONCEPT PHASE

3.1 MODELLO SVILUPPO

3.2 CONCEPT PHASE:

4. Requirements Phase

4.1 Analisi Requisiti

4.2 STAKE HOLDER

4.3 Specifica dei requisiti.

4.4 Functional Requirements

4.5 Non-Functional Requirements

4.6Requisiti di Dominio

4.7Studio Fattibilita'

4.8Analisi rischi

5.Design Phase

5.1Software Logical Structure

5.2System Architecture

5.3 Server

5.4Activity Diagram (diagramma attività)

5.5Descrizione delle GUI.

6Implementation Phase

6.1 Configuration DBMS (SERVER)

6.2Database

6.3Client

6.4 Diagramma sequenza

6.5 Security

6.6 GUI Structure

7.Testing Phase

7.1 Deployment Diagram

7.2Validation

7.3Verification

8.Release &Maintenance

8.1 Release 1.0

GLOSSARIO

1. Descrizione generale progetto

Al giorno d'oggi la realtà virtuale è molto diffusa in vari ambiti .

La realtà virtuale, per sua stessa definizione, simula la realtà effettiva. L'avanzamento delle tecnologie informatiche permette di navigare in ambientazioni fotorealistiche in tempo reale, interagendo con gli oggetti presenti in esse.

La realtà virtuale nasce dalla combinazione di dispositivi hardware e software che “collaborano” per creare uno spazio virtuale all'interno del quale l'utente può muoversi liberamente. L'accesso a questo mondo digitale è reso possibile dai visori VR e dagli accessori.

I campi di applicazione della realtà virtuale

Viste queste sue caratteristiche, la realtà virtuale ha le potenzialità per rivoluzionare – nel vero senso della parola – diversi settori dell'intrattenimento, della produzione industriale e del commercio. Il lancio del PlayStation VR, ad esempio, fornisce un assaggio di come i visori VR possono “impattare” nel settore videoludico, fornendo un'esperienza di gioco completamente differente rispetto al passato. Allo stesso modo, anche il settore turistico potrebbe mutare così profondamente che, nel giro di pochi anni, potrebbe essere irriconoscibile, ma anche il campo Scolastico stravolgendo il modo di apprendere .

Il nostro software consiste nell' implementare un sistema solare,dove l'utente farà un'esperienza in VR , oppure potrà imparare a conoscere il sistema solare ,attraverso una descrizione dettagliata e svolgere un quiz sullo stesso ,inoltre l'utente potrà acquistare determinati prodotti all'interno del market .

Il prodotto , include un are dedicata agli admin della piattaforma , dove potranno gestire l'intero sistema .

Nota :

WebVR è una specifica che rende disponibile la Virtual Reality(VR) all'interno del browser. Grazie alle API l'utilizzo di WebVR è indipendente dalla tecnologia usata come Google Cardboard, HTC Vive, Oculus Rift, Oculus Go, ecc

.

2.PANORAMICA DEL SOFTWARE.

In questo capitolo andremo a descrivere le caratteristiche che compongono il nostro software, vogliamo fare in modo che leggendo questa parte preliminare il nostro cliente oppure un qualsiasi lettore della nostra relazione riesca a capire le parti fondamentali che compongono il nostro prodotto.

2.1Linguaggi per la descrizione UML.

L'UML (Unified Modeling Language) è un linguaggio di modellazione in ambito di progettazione di software object oriented, è il linguaggio universale per modellare gli oggetti che può essere utilizzato da ogni industria produttrice di software. L'UML è, dunque, un metodo per descrivere l'architettura di un sistema in dettaglio. La forza dell'Unified Modeling Language consiste nel fatto che il processo di disegno del sistema può essere effettuata in modo tale che i clienti, gli analisti, i programmatori e chiunque altro sia coinvolto nel sistema di sviluppo possa capire ed esaminare in modo efficiente il sistema e prendere parte alla sua costruzione in modo attivo. UML è un linguaggio ricco ed esteso che può essere utilizzato per

modellare non solo l'ingegneria del software orientata agli oggetti, ma anche la struttura e il comportamento delle applicazioni e anche i processi aziendali. A causa della sua ampia portata, UML è il linguaggio visivo perfetto per comunicare informazioni dettagliate sull'architettura al maggior numero di utenti. È giusto precisare che per comprendere UML non è necessario avere una conoscenza tecnica dettagliata dell'ingegneria del software. Essendo uno **standard**, è ampiamente utilizzato e accettato come linguaggio per delineare i programmi. UML viene utilizzato in una varietà di scopi e la sua leggibilità e riutilizzabilità lo rendono una scelta ideale per i programmatori. Tra i vantaggi dell'utilizzo di UML troviamo sicuramente:

- 1) **Rappresentazione visiva:** Un diagramma UML è una rappresentazione visiva delle relazioni tra classi ed entità infatti per comprendere un software basta conoscere cosa fa ogni oggetto di una classe e come si relaziona con altri oggetti di altre classi. Grazie ad UML possiamo mostrare tutte queste informazioni in un diagramma e facilitarne la comprensione.

- 2) **Leggibilità e riusabilità:** Un diagramma UML è vantaggioso in quanto è molto leggibile. Il diagramma è pensato per essere compreso da qualsiasi tipo di programmatore e aiuta a spiegare le relazioni che ci sono in un programma in modo semplice. Tradizionalmente, per comprendere un programma, un programmatore legge direttamente il codice potrebbe trattarsi di migliaia o milioni di righe di codice in programmi molto grandi avere un diagramma UML aiuta a illustrare rapidamente queste relazioni.

Inoltre, utilizzando un diagramma per mostrare il codice in esecuzione in un programma, un programmatore è in grado di vedere il codice ridondante e riutilizzare parti di codice già esistenti piuttosto che riscrivere quelle funzioni.

3) Pianificazione: UML aiuta a pianificare un programma prima che abbia luogo la programmazione. Questo può aiutare a ridurre le spese generali durante la fase di implementazione di qualsiasi programma. Inoltre, un diagramma del modello UML è facile da modificare, mentre la riprogrammazione di una sezione di codice può essere noiosa e può richiedere molto tempo.

4) Minore spreco di risorse: utilizzando UML si vanno a ridurre i costi e i tempi di realizzazione di un prodotto.

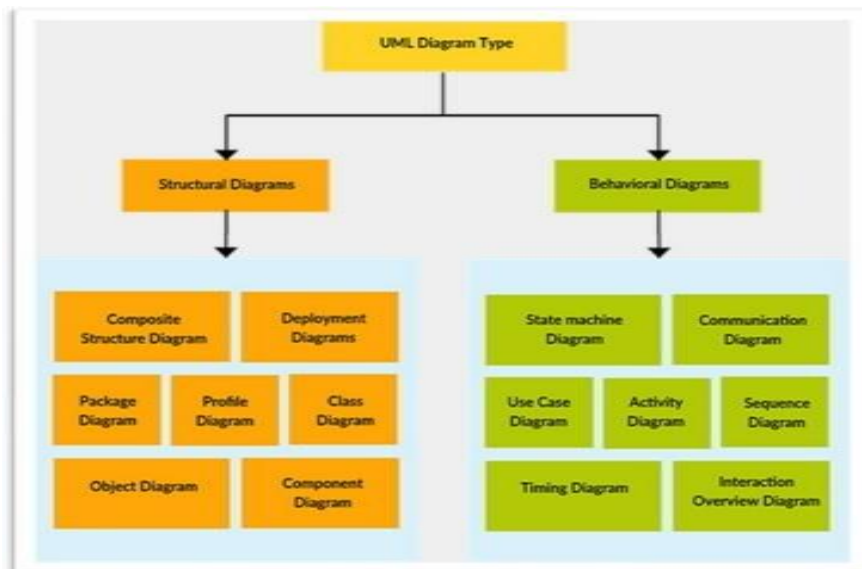


Figura 2.1

In figura (figura 2.1) è riportato un piccolo specchio di tutti quelli che sono i diagrammi UML che è possibile utilizzare.

2.3 Tecnologie implementative.

Data la natura del nostro progetto software, abbiamo deciso di utilizzare , il framework A-Frame.

A-Frame: A-Frame è molto più di un classico linguaggio di markup o di un generatore di scene 3D, A-Frame è un framework completo che si basa su entity-component per costruire applicazioni di realtà virtuale modulari ed efficienti attraverso l'uso di strutture su three.js

A-Frame supporta la maggior parte della tecnologia in suo (Vive, Rfit, Daydream, ecc..) e può essere utilizzato anche per la Realtà Aumentata grazie alla completa interazione tra utente e framework.

Lo sviluppo del software, a causa della necessità di una struttura Client Server, ha reso necessario affiancare altre tecnologie. In particolare si sceglie di utilizzare un database MYSQL.

Per editare il codice ci siamo avvalsi di IDE Atom

Per scrivere la relazione è stato utilizzato il programma di video scrittura Microsoft Word®.

Per la creazione dei diagrammi presenti nella relazione ci siamo serviti della piattaforma open source Draw.io.

Per la realizzazione dell'architettura client-server abbiamo sfruttato le tecnologie messe a disposizione da XAMPP .

2.4 Team di sviluppo.

Il team di sviluppo del software si compone di due soli elementi che svolgono il compito sia di sviluppatori che di ingegneri del software e sono:

- Francesco Cerenzia
- Alessandro Maddocco.

2.5 Requisiti team.

Per poter realizzare il prodotto software il team ha disposizione nel proprio background le seguenti skill:

- ✚ Programmazione Object Oriented, conoscenza del linguaggio JavaScript.
- ✚ Programmazione Procedurale, conoscenza linguaggio C.
- ✚ Programmazione Web, nello specifico la conoscenza dei linguaggi HTML, PHP, Javascript, CSS.
- ✚ Conoscenza e gestione dei database relazionali, utilizzo del linguaggio per la gestione degli stessi ovvero MySQL.C
- ✚ Conoscenza delle reti e dei protocolli per lo scambio d'informazioni nello specifico HTTP.
- ✚ Conoscenza del framework A-Frame

2.6Tempistiche di progetto.

Le tempistiche relative alla consegna del prodotto finito non sono state specificate, il team però prevede di consegnare il prodotto finito entro tre mesi (100 giorni circa) dall'inizio dello sviluppo.

Come descritto in precedenza il team di sviluppo si compone di soli due elementi perciò la suddivisione del lavoro non è fatta in modo rigido ma è basata su una continua collaborazione che garantisce una suddivisione equa delle cose da fare.

Trattandosi di un modello a cascata non sono richiesti prototipi , ma ci sarà un'unica iterazione finale con il cliente. Se i requisiti funzionali del prodotto soddisferanno le richieste ci sarà la release, altrimenti le tempistiche di progetto verranno aggiornate.

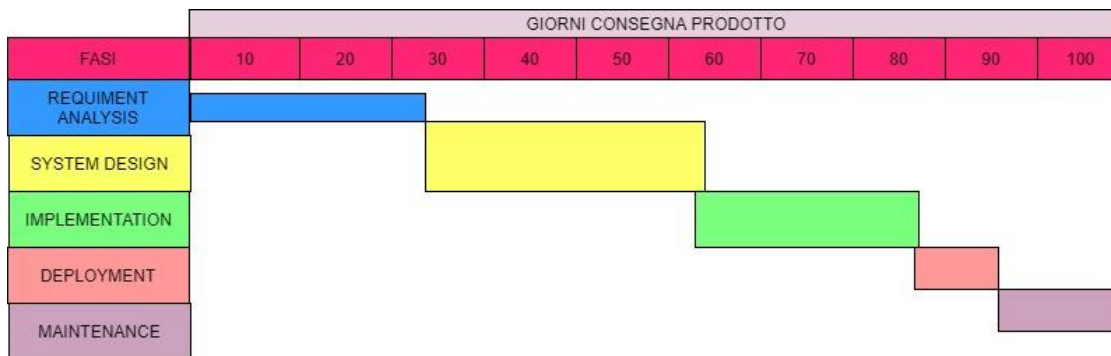


Diagramma GANTT

2.7 REQUISITI INFORMALI.

Per fare un'analisi dei requisiti informale bisogna sottoporre il cliente, ad una serie di domande per andare a identificare i requisiti da rispettare, le funzionalità da implementare e soprattutto evitare di deludere le aspettative degli stakeholder.

Le interviste che possono essere fatte si dividono in due diverse tipologie

aperta e chiusa.

Nell'intervista aperta le domande che vengono poste al nostro cliente sono delle domande aperte cioè domande a cui è possibile dare una vasta gamma di risposte.

Fatte queste considerazioni, per le caratteristiche che compongono il nostro sistema abbiamo deciso di sottoporre il nostro cliente ad un'**intervista aperta.**

Dopo un'accurata intervista fatta al cliente sono stati identificati i requisiti informali.

Partiamo descrivendo le funzionalità.

Il software dovrà permettere all'**utente** di eseguire le seguenti macro-funzionalità:

- **Gestione Accesso**
- **Utilizzo Piattaforma**
- **Gestione Piattaforma**

La *GESTIONE ACCESSO* è la funzionalità che permette di visualizzare le proprie informazioni personali e soprattutto dovrà occuparsi della creazione e accesso al profilo.

L' *UTILIZZO DELLA PIATTAFORMA* : La gestione degli utenti viene effettuata tramite un Database.

l'utente non è presente nel Database non potrà usufruire delle funzionalità del software, ma potrà visualizzare il market Una volta effettuato il login l'utente viene indirizzato nella home page del software. Dove può scegliere se :

Tour in VR ,esperienza in realtà virtuale all'interno del sistema solare(l'utente dovrà munirsi di un visore compatibile con il telefono) è presenta una chat tra utenti per discutere l'esperienza.

Leggere descrizione sistema solare, l'utente può leggere la descrizione di ogni singolo pianeta .

Quiz pianeti ,dopo aver letto la descrizione dei pianeti , l'utente potrà svolgere un quiz riguardante il sistema solare.

Market , all'interno del software è presente un market .l'utente può,visualizzare le caratteristiche del prodotto, acquistarlo,cercare un prodotto specifico, stampare la fattura.

Chatbot(assistenza): l'utente può richiedere l'assistenza , cioè , un bot fornirà un aiuto all'utente , garantendo una maggior USER-FRIENDLY

il software dovrà permettere all'**admin** la *Gestione del market* ,*Gestione piattaforma*:

Gestione del market: l'admin può gestire il market cioè , aggiungere un prodotto(in caso di esaurimento)può aggiornare i prodotti all'interno del market, ed eventualmente eliminare i prodotti contenuti all'interno del market, ed infine gestire i fornitori all del market.

Gestione piattaforma: l'admin può , gestire gli altri admin presenti nella piattaforma cioè, aggiungere un nuovo admin ,eliminare un admin presente,

aggiornare credenziali di un admin presente , può eliminare un utente ed infine gestire il quiz.

3.MODELLLO DI SVILUPPO & CONCEPT PHASE

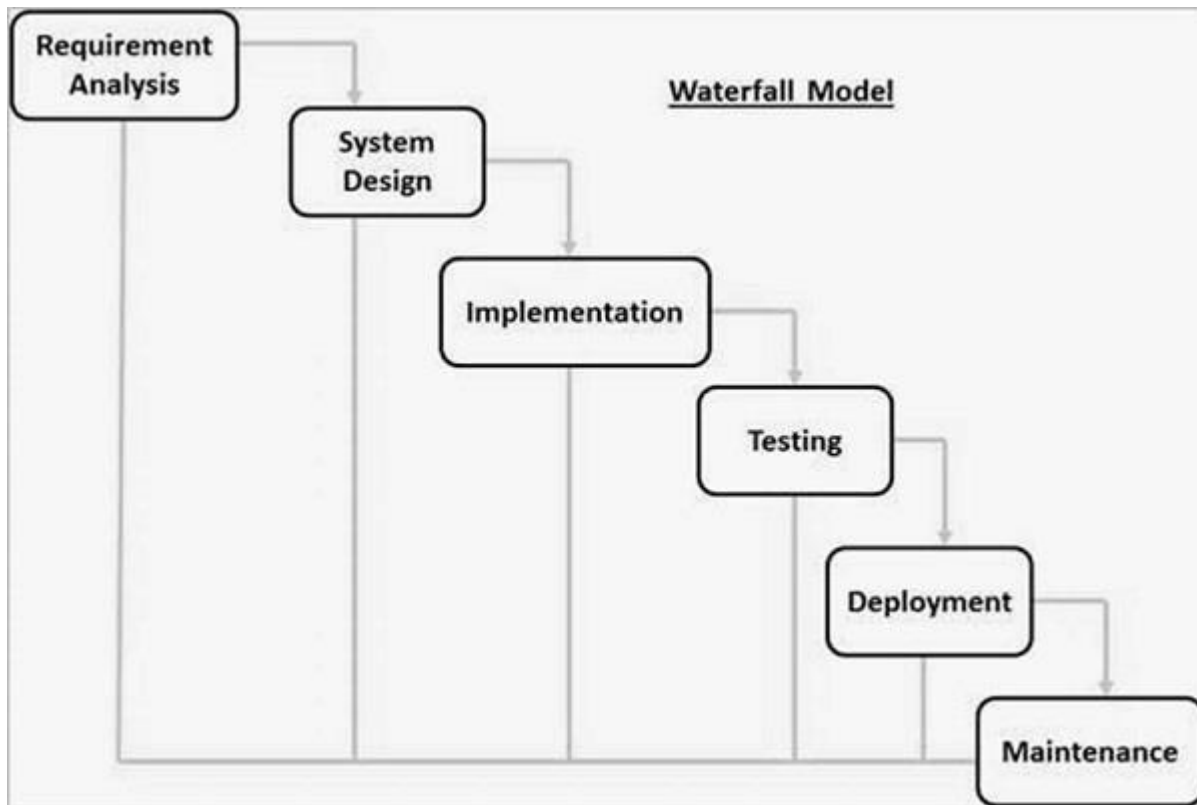
3.1MODELLO SVILUPPO

Dopo un'attenta valutazione del progetto proposto e dei requisiti presentati il team, per realizzare una WEBAPP con una WebVR integrata di questo tipo, si è scelto di adottare un approccio di tipo Plan- Driven, ed in particolare di utilizzare il modello a cascata.

È stato scelto questo approccio, perché questo modello permette di suddividere l'implementazione del sistema in più fasi distinte tra loro e occuparsi volta per volta su una singola fase.

Verranno applicate le fasi del modello a cascata così descritte:

- ❖ *Raccolta e analisi dei requisiti*
- ❖ *Design del sistema*
- ❖ *Implementazione*
- ❖ *Testing delle funzionalità*
- ❖ *Rilascio e manutenzione*



Se correttamente applicato, il metodo waterfall presenta diversi benefici nella gestione di un progetto che lo rendono preferibile rispetto ad altri approcci:

Gli sviluppatori ed il personale del cliente discutono da subito tutto ciò che dovrà essere prodotto, il che rende la progettazione e il delivery più semplice. L'enfasi sui requisiti e la progettazione prima di arrivare a produrre garantisce il minimo spreco di tempo e sforzo e riduce il rischio di slittamento dei tempi, o che le aspettative dei clienti non vengano rispettate. Poiché si discute in dettaglio degli obiettivi finali, tutti gli stakeholder acquisiscono una solida consapevolezza delle principali aspettative e dei risultati desiderati.

Il modello a cascata consente di risparmiare tempo e denaro. Viene sviluppato solo ciò che è effettivamente necessario e gli sviluppatori possono correggere eventuali errori già nelle prime fasi, permettendo alla fase di attuazione di svolgersi senza intoppi. Poiché l'obiettivo finale è chiaramente definito nelle fasi iniziali, entrambe

le parti comprendono i vincoli finanziari e temporali del progetto. Un difetto nei requisiti che non venga rilevato fino alla fase di realizzazione o alla manutenzione può costare da 50 a 100 volte di più che se fosse stato rilevato in fase di progettazione. Ciò dipende dal fatto che in fase di progettazione le risorse coinvolte sono numericamente inferiori a quelle che operano in fase di realizzazione ed esercizio ed un problema ha di conseguenza impatti diversi.

Chi opera sul progetto può facilmente determinare quanti progressi vengono fatti, dal momento che le aspettative sono chiaramente definite prima dell'inizio di un progetto. Poiché il progetto è chiaramente tracciato dall'inizio alla fine, si è in grado di misurare rapidamente i risultati e le realizzazioni rispetto a quanto dovrà essere prodotto alla fine. Il modello a cascata è ideale per squadre che lavorano a stretto contatto, in quanto offre compiti e obiettivi ben definiti e consente quindi alle persone di programmare il proprio tempo in funzione di un quadro di riferimento in cui sono chiare le interdipendenze tra il lavoro dei vari team.

Il modello a cascata consente un maggior livello di controllo sulle fasi di realizzazione e sui rischi associati proprio perché conserva una visione integrata di tutto ciò che deve essere fatto e delle interrelazioni. Su progetti complessi oppure su progetti innovativi il metodo waterfall consente di presidiare meglio le possibili criticità. Sarebbe quindi opportuno integrare i cicli di sviluppo agile all'interno di un modello a cascata per ottenere maggior solidità in termini di controlli.

3.2 CONCEPT PHASE:

In questa fase vengono definite le linee guida e le caratteristiche principali del progetto. Sulla base della descrizione delle caratteristiche del progetto, si effettua un'analisi del dominio applicativo, uno studio di fattibilità.

Feasibility Study

Prima di procedere con lo sviluppo del software, è necessario affrontare uno studio di fattibilità per valutare sia i possibili risultati sia la necessità di rispettare le scadenze previste, oltre che i costi.

Bisogna che il team analizzi delle aree di fattibilità che afferiscono alla fattibilità del progetto. Nello specifico viene analizzata:

- ❖ *Fattibilità tecnica*
- ❖ *Fattibilità giuridica*
- ❖ *Fattibilità operativa*
- ❖ *Tempo della fattibilità*
- ❖ *Fattibilità economica*
- ❖ Technical Faesability

Il team è in grado di determinare le risorse tecniche e convertire le idee del cliente in sistemi operativi funzionanti, in quanto possiede le nozioni e gli strumenti per operare al meglio.

Le risorse tecniche vengono soddisfatte, dal momento che il team possiede tutti gli strumenti tecnici ed è a conoscenza delle soluzioni algoritmiche e architetturali per la corretta realizzazione del software.

Inoltre, l'elevata conoscenza del dominio applicativo garantisce una posizione favorevole per le scelte progettuali e dunque ogni stakeholder verrà soddisfatto.

❖ **Legal Faesability**

In nessun modo, gli aspetti del progetto sono in conflitto con requisiti legali né tantomeno con le vigenti normative, che ne regolano inoltre la protezione dei dati.

❖ **Operational Faesability**

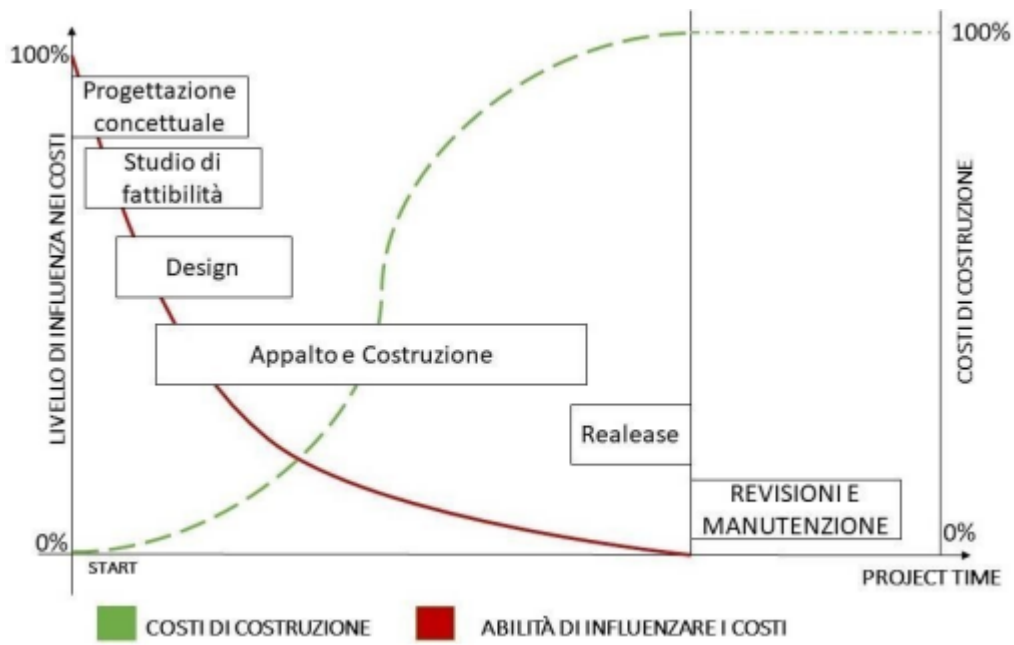
Il team ha già individuato una metodologia operativa, per portare al termine la realizzazione software. Utilizzando uno sviluppo modulare del sistema (modularizzazione) vengono evitati problemi all'interno del team, oltre che favorire l'integrazione dei diversi componenti e la loro eventuale modifica.

❖ **Faesability Time**

Il team di lavoro, avendo già operato congiuntamente allo sviluppo di software analoghi, possiede la giusta padronanza delle tempistiche e traendo benefici da queste esperienze si ritiene fermamente sicuro che i tempi per la realizzazione sono più che adeguati.

❖ **Economic Faesability**

La valutazione economica per la realizzazione del software, mette il team in una posizione vantaggiosa per lo sviluppo, in quanto i costi richiesti sono esigui e dunque il rapporto costi/benefici viene fortemente influenzato dai benefici prodotti dalla realizzazione del software, e per determinare ciò il team ha attentamente valutato l'analisi di una ricerca di mercato.



4Requirements Phase

4.1Analisi Requisiti

I requisiti sono caratteristiche, proprietà e comportamenti che l'applicazione deve avere al termine dello sviluppo. Per effettuare l'analisi dei requisiti si individuano i casi d'uso, gli attori e le relazioni che intercorrono tra di essi. I casi d'uso descrivono le interazioni tra il sistema e l'utente, mentre gli attori rappresentano i ruoli degli utenti che interagiscono con il sistema. Essi possono anche essere dei sottosistemi. Un attore svolge uno o più casi d'uso e allo stesso modo un caso d'uso coinvolge uno o più attori.

4.2STAKE HOLDER

- Committente: colui che commissiona la realizzazione del software
- Utente : tipologia di utenti che usufruiscono del servizio , possono registrarsi al sito in maniera gratuita , autenticarsi, apprendere sul sistema solare , svolgendo attività come quiz , tour in realtà virtuale o acquistare prodotti .
- Admin: tipologia di utenti che gestiscono la piattaforma, hanno il compito di garantire la giusta correttezza della piattaforma. Devono garantire sempre la reperibilità di un prodotto, eliminare un utente se necessario, aggiornare le domande, gestire i fornitori .

4.3 Specifica dei requisiti.

Per specifica dei requisiti si intende una descrizione completa del comportamento di un sistema software da sviluppare. Per requisito si intende qualcosa che esso deve fare o per meglio dire una caratteristica che esso deve possedere. Nella SE si diversificano i tipi di requisiti infatti abbiamo requisiti funzionali e requisiti non funzionali.

4.4 Functional Requirements

Riassumendo, è possibile individuare i seguenti requisiti funzionali, ovvero quei requisiti che descrivono cosa il sistema deve fare (le sue funzionalità), in termini delle azioni che il prodotto deve svolgere.

RQ1.0: GESTIONE ACCESSO(UTENTE)

RQ1.0.1: Registrazione utente

RQ1.0.2: Login

RQ1.0.3: Logout

RQ1.1: GESTIONE ACCESSO (ADMIN)

RQ1.1.1: Login

RQ1.1.2: Logout

RQ1.1.2: Profilo

RQ2.0: UTILIZZO PIATTAFORMA(UTENTE)

RQ2.0.1: Tour Vr(chat compresa)

RQ2.0.2: Market(compra / visualizza / stampa)

RQ2.0.3: Descrizione pianeti

RQ2.0.4: Quiz sui pianeti

RQ2.0.5: Chatbot

RQ3: GESTIONE PIATTAFORMA(ADMIN)

RQ3.1: Elimina utente

RQ3.2: Inserisci admin

RQ3.3: Elimina admin

RQ3.4: Aggiorna admin

RQ3.5: Aggiungi prodotto

RQ3.5: Modifica prodotto

RQ3.5: Elimina prodotto

RQ3.6:Aggiungi Fornitori

RQ3.7:Aggiorna Fornitori

RQ3.8:Elimina Fornitori

RQ3.9: Aggiungi domande/risposte al quiz

RQ3.10:Elimina Domande quiz

4.5Non-Functional Requirements

I requisiti non funzionali descrivono le qualità del software e i vincoli che esso deve rispettare.

RQ-NF1: Compatibilità con diverse piattaforme

RQ-NF2: Interfaccia GUI user-friendly

RQNF3: Utilizzo di una database per la memorizzazione dei dati

RQNF4: Basso carico d'elaborazione da parte del dispositivo

RQ-NF5:Le task che vengono svolte dal software devono essere svolte in un tempo ragionevole.

RQ-NF6:Il software deve essere scalabile, ovvero permettere l'aggiunta di

funzionalità e ulteriori requisiti a posteriori senza grandi stravolgimenti del codice, ma solo con l'aggiunta della relativa porzione.

RQ-NF7:Un requisito è l'usabilità dell'applicativo. In questo caso, l'applicazione non svolge compiti complessi o computazionalmente pesanti, ma, dato l'alto numero di micro-operazioni che l'applicazione è chiamata a svolgere, è importante fare in modo che l'interfaccia sia bene organizzata

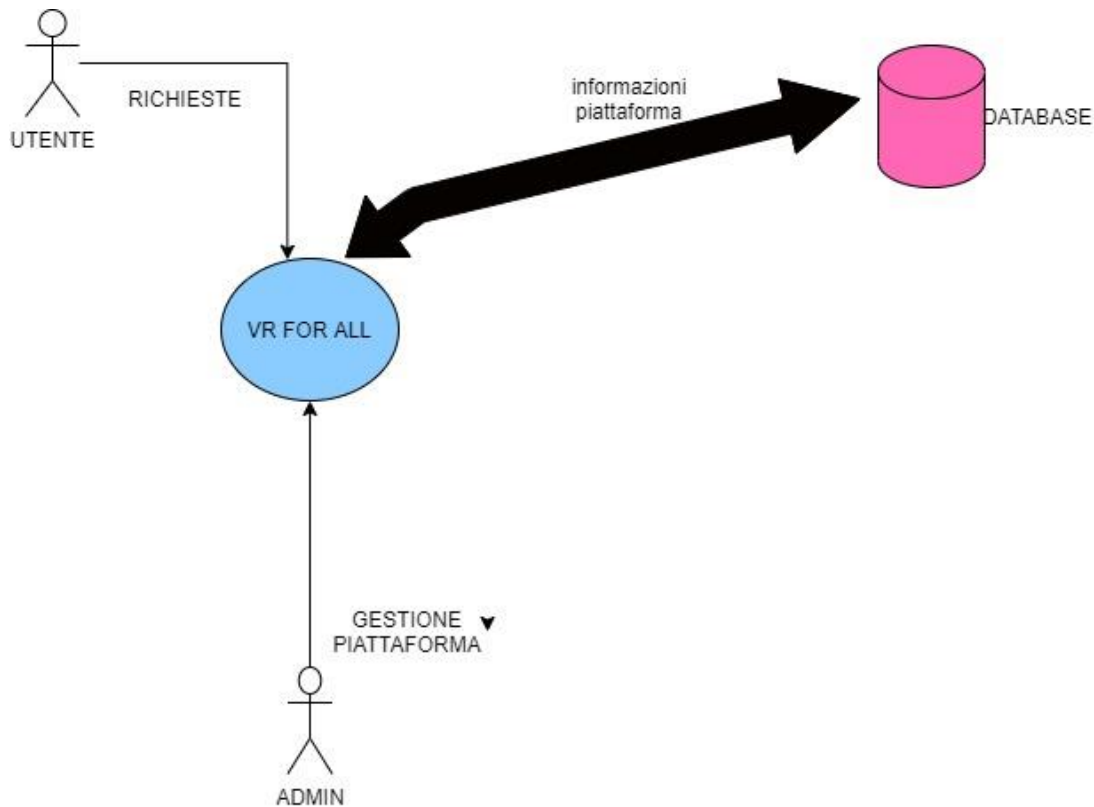
4.6REQUISITI DI DOMINI

Il sistema deve poter essere utilizzato in ambienti con sistema operativo Windows.

Application Domain

Il dominio applicativo del software appartiene al mondo delle webApplication e, nello specifico, da la possibilità alle persone che utilizzano la piattaforma la possibilità di apprendere nozioni sul sistema solare , integrando anche una realtà aumentata .Dal momento che il team conosce a pieno questo ambito, non è necessario richiedere l'ausilio di esperti del dominio.

Per descrivere al meglio tale dominio applicativo si fa uso di un Context Diagram. In particolare, si distinguono due macro tipologie di sottodomini: un **machine domain** e un insieme di **problem domain**. Questi ultimi a loro volta vengono suddivisi in **designed domain** (che esprimono la possibilità, in fase di realizzazione della soluzione, di progettare le caratteristiche) e **given do- main** (non è possibile intervenire su nessuna caratteristica di dominio).



CONTEXT DIAGRAM

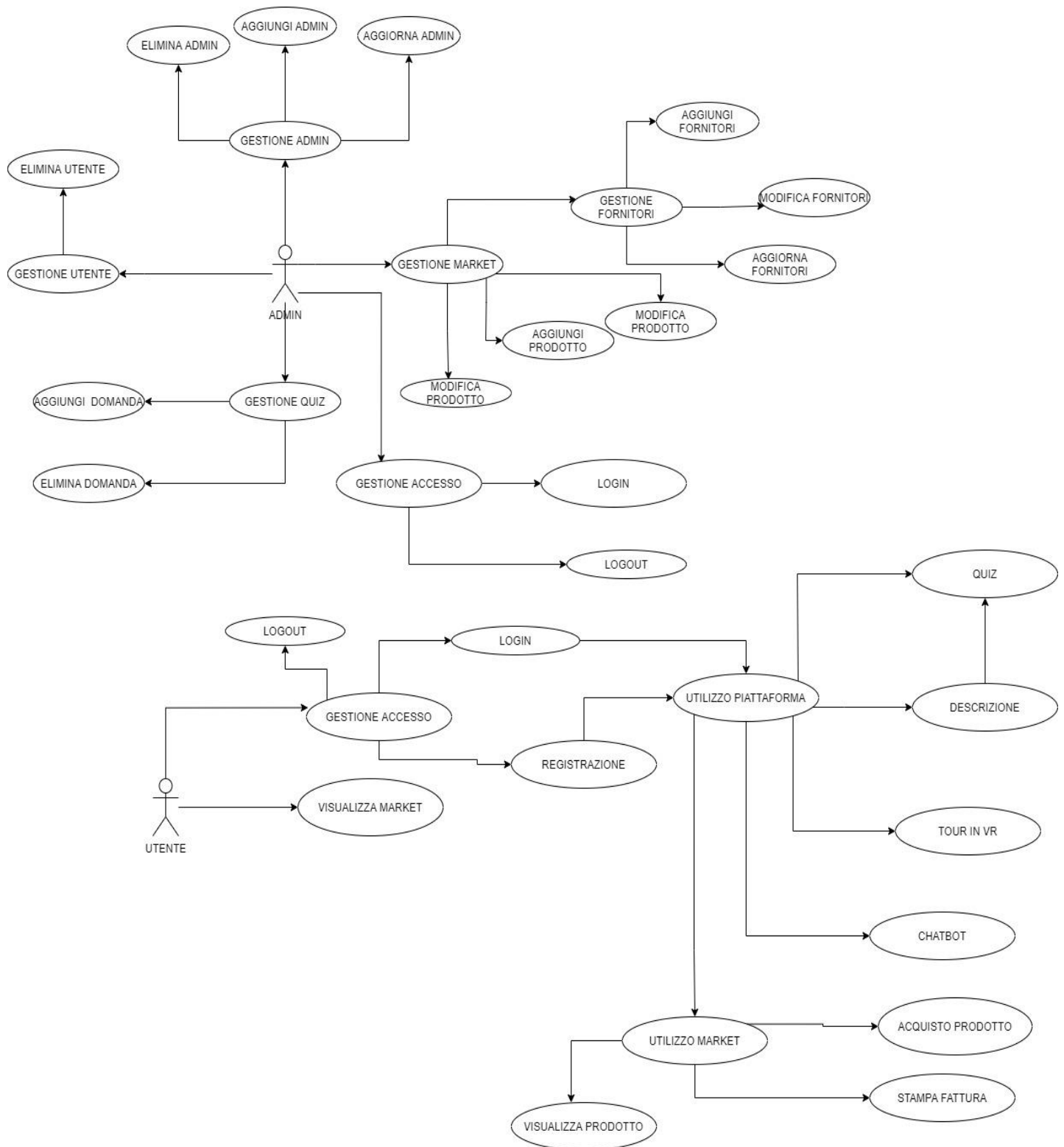
1. **VR FOR ALL**(*machine domain*): è il sistema da sviluppare
2. **UTENTI**(*given domain*): rappresenta gli utenti che useranno il programma
3. **RICHIESTE**(*designed domain*): gli utenti faranno delle richieste per ordinare i prodotti sul market.
4. **ADMIN**(*given domai*) : altra tipologia di utenti che gestiscono la piattaforma
5. **GESTIONE PIATTAFORMA**(*given domain*): gli admin hanno il compito di gestire la piattaforma dunque , gestiscono admin, utenti e market .

Per modellare i requisiti del sistema si utilizza il diagramma dei casi d'uso dell'UML, che permette di individuare chi o che cosa ha a che fare con il sistema (**attore**) e che cosa l'attore può fare (**caso d'uso**)

DIAGRAMMA CASI D'USO

Legenda:

Simbolo	Significato
Uomo stilizzato	Attore
Ellisse	Caso d'uso
Freccia grande	Generalizzazione
Frecce rosse	Inclusione o estensione



Dallo schema è possibile andare ad identificare le funzioni più importanti che il nostro software dovrà offrire.

Tipologia utente:

- + REGISTRAZIONE
- + LOGIN
- + TOUR IN VR
- + CHAT
- + CHATBOT(ASSISTENZA)
- + DESCRIZIONE PIANETI
- + QUIZ
- + VISUALIZZA PRODOTTO
- + ORDINA PRODOTTO
- + STAMPA FATTURA

Tipologia Admin:

- + LOGIN
- + ELIMINA UTENTE
- + AGGIUNGI DOMANDE
- + ELIMINA DOMANDE
- + AGGIUNGI PRODOTTO
- + ELIMINA PRODOTTO
- + AGGIORNA PRODOTTO
- + AGGIUNGI ADMIN
- + ELIMINA ADMIN
- + AGGOIRNA CREDENZIALI ADMIN

SCENARI CASI D'USO:

Primo caso d'uso: registrazione

Attore: utente

Precondizione: L'utente non è registrato al sistema e non può usare gli strumenti forniti dal software.

Post condizione: una volta finita la registrazione senza errori, l'utente potrà effettuare il login e usufruire del software.

Scenario principale:

L'utente va nella selezione relativa alla registrazione.

Inserisce un nome utente, un'email e una password ripetuta due volte.

Effettua la registrazione tramite il relativo pulsante.

L'utente viene aggiunto al sistema.

Scenario alternativo:

Nella fase di inserimento le credenziali sono state errate

In questi casi la registrazione fallisce mostrando a schermo l'errore.

L'utente può rifare la registrazione reinserendo di nuovo informazioni necessarie.

Secondo caso d'uso: login

Attore: utente

Precondizione: L'utente è registrato al sistema per poter usare i suoi strumenti deve accedere al tool.

Post condizione: Una volta acceduto al sistema con la fase di login, l'utente può usare le funzionalità offerte dal sistema.

Scenario principale:

L'utente va nella selezione relativa al login.

Inserisce lo username e la password usate per la registrazione.

Effettua il login tramite il relativo pulsante.

L'utente accede al sistema.

Scenario alternativo:

Nella fase di inserimento delle credenziali username potrebbero non corrispondere a nessun utente registrato al sistema.

Viene mostrato l'errore a schermo e il login fallisce.

L'utente può riprovare ad effettuare il login.

Terzo caso d'uso: Descrizione dettagliata del sistema solare .

Attore: utente

Precondizione : L'utente è registrato al sistema per poter usare i suoi strumenti deve accedere al tool.

SCENARIO PRINCIPALE:

All'interno di questa sezione l'utente trova una sezione dedicata alla descrizione dettagliata del sistema solare. L'utente selezionando il tasto DESCRIPTION viene portato nella pagina desiderata. All'interno troverà la descrizione dettagliata dei singoli pianeti e la possibilità di fare il quiz

SCENARIO ALTERNATIVO:

L'utente ha una connessione non stabile, potrebbe non utilizzare in maniera corretta la piattaforma

QUARTO CASO :Descrizione del tour in VR

Attore : Utente

Precondizione : l'utente ha effettuato il login

Post-condizione: l'utente utilizza il servizio

SCENARIO PRINCIPALE :

All'interno di questa pagina l'utente avrà la possibilità di vivere un'esperienza di in VR riguardante il sistema solare. L'utente navigherà tra i pianeti del sistema solare.

L'utente tramite il VR (preferibile un Google Cardboard VR per l'utilizzo del software) interagisce con il software

SCENARIO ALTERNATIVO:

Cattiva connessione ; il software non viene utilizzato correttamente

Problema di connessione tra telefono e PC.

QUINTO SCENARIO :QUIZ

Attore: utente

Precondizioni : L'utente deve aver fatto l'accesso al sito

Postcondizioni : L'utente può usare il servizio

SCENARIO PRINCIPALE :

L'utente può svolgere un quiz su ciò che ha letto nelle descrizioni dei pianeti

L'utente può svolgere un quiz sul sistema solare.

SCENARIO ALTERNATIVO:

Cattiva connessione ; il software non viene utilizzato correttamente

QUINTO SCENARIO :CHATBOT

Attore: utente

Pre_condizioni: L'utente deve aver fatto l'accesso al sito

Post-condizioni: L'utente utilizza il servizio

SCENARIO PRINCIPALE

L'utente può scegliere questa funzionalità per essere assistito durante l'utilizzo del software, cioè viene indirizzato alle varie funzionalità messe a disposizione

SCENARIO ALTERNATIVO:

Cattiva connessione ; il software non viene utilizzato correttamenteSCENARIO

SESTO SCENARIO :VISUALIZZA MARKET

Attore:utente

Pre_condizioni: L'utente deve aver fatto l'accesso al sito

Post-condizioni: L'utente utilizza il servizio

SCENARIO PRINCIPALE:

L'utente può visualizzare i prodotti nel dettaglio , ed eventualmente comprare il prodotto, per agevolare l'utente il software mette a disposizione una barra di ricerca all'interno del market.

SCENARIO ALTERNATIVO:

Cattiva connessione ; il software non viene utilizzato correttamenteSCENARIO

SETTIMO SCENARIO :ORDINA PRODOTTO

Attore:utente

Pre_condizioni: L'utente deve aver fatto l'accesso al sito

Post-condizioni: L'utente utilizza il servizio

SCENARIO PRINCIPALE:

L'utente dopo aver visualizzato i prodotti disponibili , può acquistarli , inserendone la quantità desiderata , indirizzo e il fornitore presente nella zona.

SCENARIO ALTERNATIVO:

Cattiva connessione ; il software non viene utilizzato correttamenteSCENARIO

OTTAVO SCENARIO :STAMPA FATTURA

Attore:utente

Pre_condizioni: L'utente deve aver fatto l'accesso al sito

Post-condizioni: L'utente utilizza il servizio

SCENARIO PRINCIPALE:

L'utente può decidere di stampare la fattura , dopo aver acquistato un determinato prodotto .

SCENARIO ALTERNATIVO:

Cattiva connessione ; il software non viene utilizzato correttamenteSCENARIO

Primo caso d'uso: login

Attore:admin

Precondizione: I dati dell'admin sono presenti nel database.

Postcondizione: Una volta effettuato il login l'admin si troverà nell'homepage dedicata.

Scenario principale:

1.L'admin inserendo le credenziali correttamente accede all'area dedicata agli utenti.

Scenario alternativo:

1.L'admin inserisce le crdenziali in maniera errata.

2.Viene visualizzato un messaggio di errore

3.L'admin inserisce nuovamente le proprie credenziali.

Secondo caso d'uso: Aggiungi prodotto

Attore:admin

Precondizione: L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin aggiunge il prodotto correttamente

Scenario principale:

1.L'admin recandosi nella sezione, aggiunge un nuovo prodotto nel market, con le opportune specifiche tecniche.

Scenario alternativo:

1.La connessione è instabile

2. Vi è un malfunzionamento della piattaforma
3. L'admin riproverà ad aggiungere nuovamente il prodotto

Terzo caso d'uso: Modifica prodotto

Attore: Admin **Precondizione:** L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin modifica il prodotto correttamente

Scenario principale:

1. L'admin recandosi nella sezione apposita, modifica un prodotto.

Scenario secondario:

1. La connessione è instabile
2. Vi è un malfunzionamento della piattaforma
3. L'admin riproverà a modificare nuovamente il prodotto

Quarto caso d'uso: Elimina prodotto

Attore: Admin **Precondizione:** L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin elimina il prodotto correttamente

Scenario principale:

1. L'admin recandosi nella sezione apposita, elimina un prodotto.

Scenario alternativo:

1. La connessione è instabile
2. Vi è un malfunzionamento della piattaforma

3.L'admin riproverà ad eliminare nuovamente il prodotto

Quinto caso d'uso: Elimina domanda

Attore:Admin

Precondizione: L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin elimina la domanda correttamente

Scenario principale:

1.L'admin recandosi nella sezione apposita, elimina una domanda.

Scenario secondario:

1.La connessione è instabile

2.Vi è un malfunzionamento della piattaforma

3.L'admin riproverà ad eliminare nuovamente la domanda

Sesto caso d'uso: Aggiunge domanda

Attore: Admin

Precondizione: L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin aggiunge la domanda correttamente

Scenario principale:

1.L'admin si reca nella sezione apposita.

2.L'admin inserisce il titolo della domanda.

- 3.L'admin inserisce le risposte.
- 4.L'admin sceglie quale risposta sia corretta.
- 5.L'admin aggiunge una nuova domanda correttamente.

Scenario alternativo:

- 1.La connessione è instabile
- 2.Vi è un malfunzionamento della piattaforma
- 3.L'admin riproverà ad aggiungere nuovamente la domanda

Settimo caso d'uso: Aggiungi ADMIN

Attore:admin

Precondizione: L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin aggiunge il prodotto correttamente

Scenario principale:

- 1.L'admin recandosi nella sezione, aggiunge un admin , con le opportune specifiche tecniche.

Scenario alternativo:

- 1.La connessione è instabile
- 2.Vi è un malfunzionamento della piattaforma
- 3.L'admin riproverà ad aggiungere nuovamente l'admin

Ottavo caso d'uso: Modifica admin

Attore: Admin

Precondizione: L'admin effettua l'accesso alla piattaforma. **Postcondizione:** L'admin modifica il prodotto correttamente

Scenario principale:

1.L'admin recandosi nella sezione apposita, modifica un admin esistente .

Scenario alterntivo:

- 1.La connessione è instabile
- 2.Vi è un malfunzionamento della piattaforma
- 3.L'admin riproverà a modificare nuovamente il prodotto

Nono caso d'uso: Elimina adim

Attore: Admin **Precondizione:** L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin elimina il prodotto correttamente

Scenario principale:

- 1.L'admin recandosi nella sezione apposita, elimina un admin presente .

Scenario alternativo:

- 1.La connessione è instabile
- 2.Vi è un malfunzionamento della piattaforma
- 3.L'admin riproverà ad eliminare nuovamente il prodotto

Decimo caso d'uso: Aggiungi fornitore

Attore:admin

Precondizione: L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin aggiunge il prodotto correttamente

Scenario principale:

1.L'admin recandosi nella sezione, aggiunge un nuovo fornitore del market, con le opportune specifiche tecniche.

Scenario alternativo:

1.La connessione è instabile

2.Vi è un malfunzionamento della piattaforma

3.L'admin riproverà ad aggiungere nuovamente il prodotto

Undicesimo caso d'uso: Modifica fornitore

Attore: Admin **Precondizione:** L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin modifica il prodotto correttamente

Scenario principale:

1.L'admin recandosi nella sezione apposita, modifica un fornitore esistente

Scenario secondario:

1.La connessione è instabile

2.Vi è un malfunzionamento della piattaforma

3.L'admin riproverà a modificare nuovamente il prodotto

Dodicesimo caso d'uso: Elimina fornitore

Attore: Admin **Precondizione:** L'admin effettua l'accesso alla piattaforma.

Postcondizione: L'admin elimina il prodotto correttamente

Scenario principale:

1.L'admin recandosi nella sezione apposita, elimina un fornitore esistente

Scenario alternativo:

1.La connessione è instabile

2.Vi è un malfunzionamento della piattaforma

3.L'admin riproverà ad eliminare nuovamente il prodotto

4.8 Studio fattibilità

Il committente del software ha proposto il suo budget per il progetto, le tempistiche di realizzazione ed i requisiti del sistema. Tali informazioni sono scaturite tramite delle interviste con il committente stesso e dopo una fase molto attenta e di negoziazione sul compenso finale, sulle date di scadenza e sui requisiti, si è fatta la scelta di accettare il progetto proposto.

Il cliente si è dimostrato molto aperto e capace nella descrizione delle specifiche rendendo l'operazione molto semplice e rapida. Inoltre dallo studio del problema è stato possibile capire che il sistema è realizzabile da un team, non troppo grande, che ha conoscenze sulla programmazione, ed in particolare su quella ad oggetti e sia familiare alla creazione di una interfaccia grafica.

Si noti che per l'implementazione del progetto il team ha acquisito o deve acquisire competenze riguardo lo sviluppo di una WebVR

4.9 Risk Analysis

All'interno del processo di sviluppo del software ci sono molte aree di rischio ed il team dovrà sviluppare dei modi per assicurare che una possibile occorrenza di un evento indesiderato possa essere individuata, corretta e risolta per evitare conseguenze disastrose.

In relazione al software dobbiamo distinguere due diversi casi possibili di rischi:

- ✚ Rischi di progetto

- ✚ Rischi di prodotto

Rischi di Progetto

Il rischio principale che è necessario evitare durante la progettazione del software è quello di non riuscire a rendere il prodotto adattabile ad ogni tipologia di utente.

Nel nostro caso , l'utente senza un visore non potrebbe usare una funzionalità del sistema.

Per evitare di cadere in questo problema, si sceglie di andare a delineare molteplici opzioni per rendere il software universalmente funzionale.

Un rischio da non sottovalutare nello sviluppo di un software è quello di non riuscire a consegnare il **prodotto** entro e non oltre le tempistiche pattuite, ma il team tramite le proprie competenze e la scelta di modularizzare lo stesso, potrà aumentare la produttività in quanto lavorando separatamente sui singoli componenti, il consolidamento finale sarà semplice e ben strutturato e nel caso di futuri update del software, le modifiche saranno anch'esse modularizzate (ottimizzandone lo sviluppo) . La progettazione utilizzando la metodologia ad incremento permette di avere sin da subito chiare le specifiche del software, dunque non sussiste il rischio che i requisiti funzionali possano evolvere in modo tale da stravolgere un prototipo ed a giovarne è il team di sviluppo.

Rischi di Prodotto

I rischi di prodotto che si possono trovare andando a sviluppare il software sono molteplici, ad esempio l'incapacità di potersi adattare alle piattaforme su cui viene impiantato. Come già ampiamente descritto, le tecnologie scelte risolvono questo possibile problema in fase di deployment, accertando che non sia un prodotto fortemente agganciato ad una determinata piattaforma.


Il Team, grazie alla propria esperienza consolidata nel tempo, è a conoscenza dei requisiti per rendere un software appetibile all'utente finale, e dunque lo sviluppo delle interfacce e di conseguenza dello sviluppo dei prototipi non è casuale ma certamente mirata in modo tale che non possa esserci il rischio che un errore di "costruzione" delle interfacce possa determinare la non consegna o il ritardo del prodotto finale. La metodologia della prototipizzazione sicuramente avvantaggia il team a rispettare le scadenze e i requisiti funzionali scelti dal cliente.

RISCHI	VALUTAZIONE RISCHI O	PROBABILITA' CHE ACCADA	CONSEGUENZE	METODO DI RISOLUZIONE
Adattabilità software ad	Alto.	Poco probabile.	Utente che non riesce ad	Risulta molto difficile che un utente non riesc

ogni utente.			utilizzare il software.	ad utilizzare il software in quanto il nostro sistema permette a chiunque di utilizzarlo.
Assenza di prodotti nel market.	Alto.	Probabile.	Utente che attende troppo tempo l'aggiunta del prodotto ricercato.	Sarà compito dell'admin tenere il market sempre aggiornato, e con prodotti sempre reperibili
Poca voglia di usare il sistema da parte dell'utente	Medio.	Poco Probabile.	L'utente può stancarsi della piattaforma.	Il nostro sistema è composto da diverse funzionalità per mantenere il cliente sempre stimolato.
L'utente non	Alto.	Probabile.	L'utente	Questo non dipende

sistema solare			la propria curiosità sul sistema solare.	nostro software è progettato per qualsiasi età degli utenti, pertanto garantiamo di invogliare l'utente con varie funzionalità di intrattenimento.
Market che non soddisfa i gusti del cliente	Medio.	Probabile.	Un utente può ritenere il market poco fornito	Tramite i feedback degli utenti il market avrà prodotti nuovi e aggiornati.

Richieste del cliente poco chiare.	Medio.	Probabile.	Il team non riesce a rispettare i requisiti.	Avendo fatto un'intervista e poi scelto una metodologia prototipale il team è certamente sicuro che i requisiti verranno rispettati.
------------------------------------	--------	------------	--	--

Prodotto non funzionante su una piattaforma.	Basso.	Poco Probabile.	Il sistema non riesce a funzionare su uno o più device con piattaforme diverse.	Avendo utilizzato tecnologie nuove e basate sul web, il problema non sussiste
Allenamento che va a ledere la salute del nostro utente.	Basso.	Poco Probabile.	Un nostro utente ha problemi di salute derivati dall'allenamento.	Avendo implementato un meccanismo di controllo sulla salute degli utenti (questionario) e avendo utilizzato studi condotti per sei anni da personale specializzato per ottenere le schede di allenamento da svolgere, avendo un team di medici a disposizione degli utenti, il rischio è molto basso quasi inesistente.
Impossibilità di misurare il proprio corpo.	Basso.	Poco Probabile.	Un utente non riesce a misurare i propri muscoli e quindi non può utilizzare i servizi offerti.	All'interno del nostro sistema sono presenti diversi consigli su come misurare i muscoli. Dal punto di vista logistico è molto semplice, basta usare un metro da sarta  v.
Tempistiche di consegna non adeguate.	Basso.	Poco Probabile.	Il prodotto non viene consegnato al cliente nei tempi stabiliti.	Lo studio di fattibilità condotto in precedenza e le metodologie di sviluppo software usate dal team fanno in modo che sia impossibile la non-consegna del prodotto finito

Evoluzione	Medio.	Probabile.	Il cliente trova	Utilizzando un approccio Plan-driven a cascata è
delle richieste cliente.			il sistema non congruo a quelle che erano le richieste iniziali.	possibile anche in corso d'opera, apportare modifiche al sistema andando ad avallare le richieste del cliente.
Conflitti all'interno del team.	Basso.	Improbabile.	Il team non va d'accordo e non riesce a portare avanti lo sviluppo del software.	I pretesi ambighiati che i clienti non possono interferiscono tra di loro, ha già esperienze passate di lavoro in comune quindi il rischio è inesistente.
Impossibilità nel realizzare il sistema.	Basso.	Improbabile.	Il team non riesce a realizzare il sistema.	Il team ha fatto un attento studio di fattibilità e oltre a questo possiede ottime skill che permetteranno di realizzare il sistema senza problemi e interruzioni.

5Design Phase

Per la fase di progettazione del prototipo si utilizza una filosofia top-down. Si definiscono quindi dapprima le componenti principali, specificando la loro interazione, andando poi ad approfondirle nel dettaglio in seguito.

5.1Software Logical Structure

Come si può notare già con l'analisi dei requisiti, a livello logico il software può essere suddiviso nelle seguenti parti:

1. GESTIONE ACCESSO
2. UTILIZZO PIATTFORMA
3. GESTIONE PIATTAFORMA

A questo punto può risultare utile descrivere in modo più o meno dettagliato le funzionalità delle diverse parti, in modo da dare una visione completa dell'interazione dei vari elementi al programmatore.

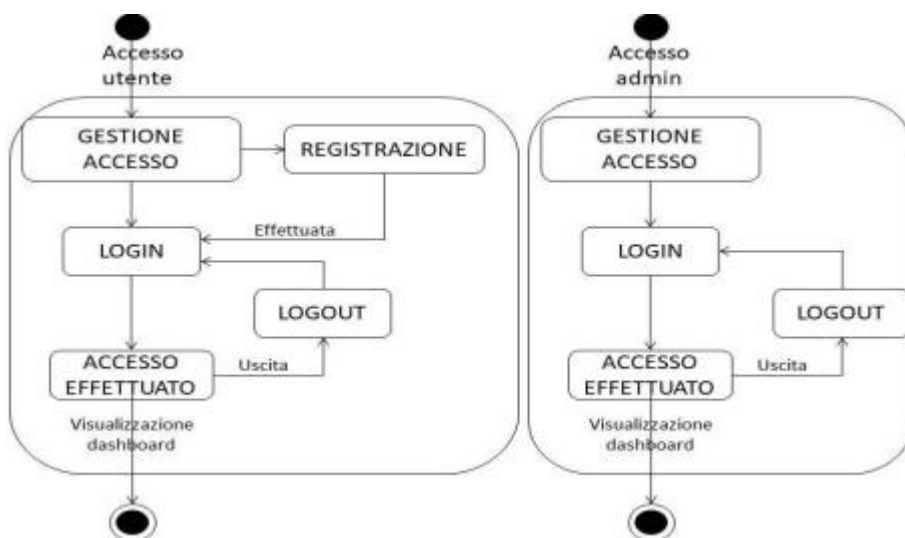
A tal fine si fa uso di alcuni State Diagram, che permettono di dare un'idea chiara del funzionamento logico che si prevede debba avere il programma finale in funzione degli utenti

GESTIONEACCESSO

Una volta che un utente qualsiasi effettua l'accesso alla piattaforma dovrà identificarsi nella tipologia che lo contraddistingue, per la quale le sue credenziali funzionano.

Scelta la tipologia gli verranno presentate alcune opzioni, nel caso dell'**admin** o **e** sarà presente esclusivamente il login, nel caso si identifichi come un **utente** avrà la possibilità di effettuare la registrazione alla piattaforma ma nel caso fosse già registrato usufruirà del login ed accedere.

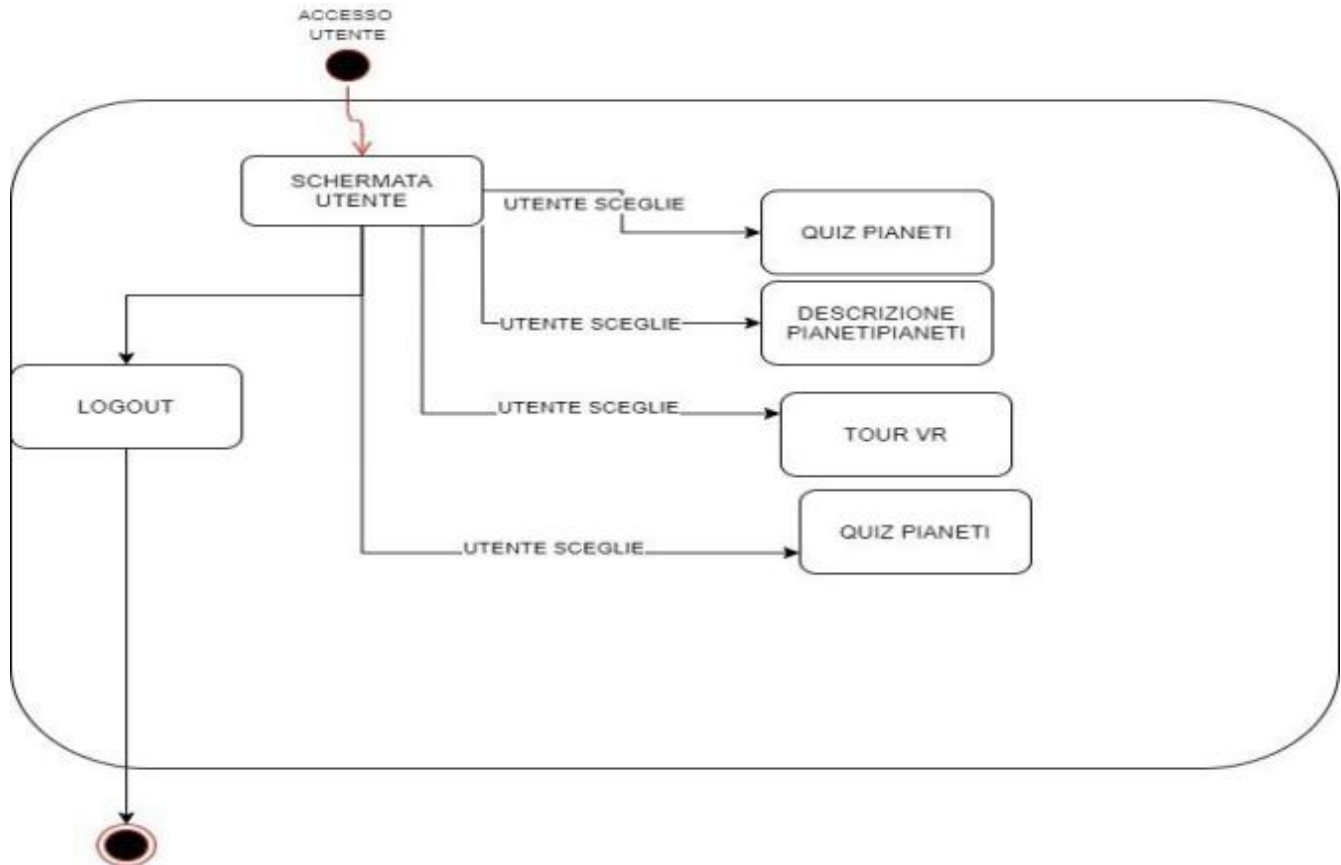
Una volta effettuato l'accesso si potrà sempre effettuare il logout ed uscire dalla piattaforma.



State diagrams: Gestione accesso in funzione del utente

UTILIZZO PIATTAFORMA (UTENTE)

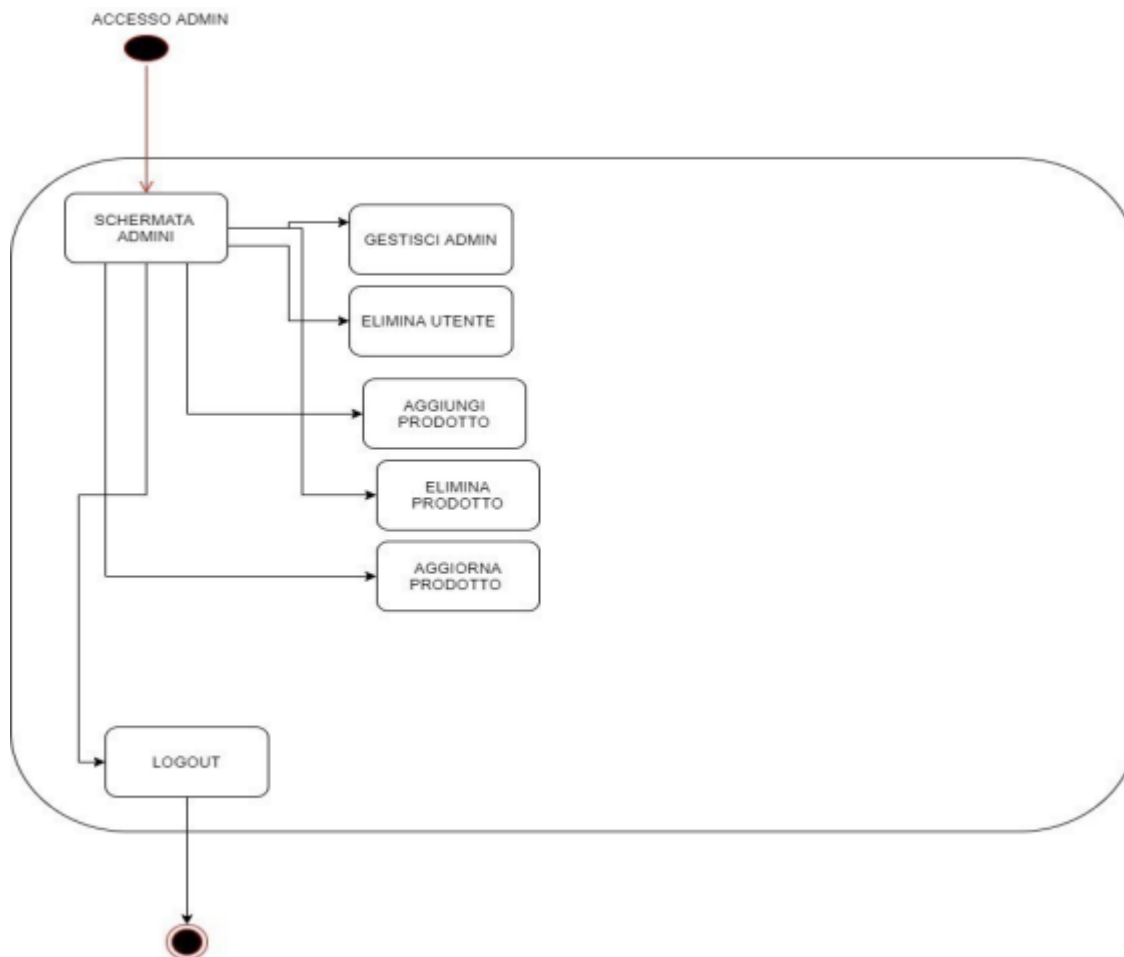
Dopo aver effettuato l'accesso l'utente ,può utilizzare il software e decidere quale operazione fare tra *QUIZ SUI PIANETI*, *DESCRIZIONE SUI PIANETI*, *TOUR IN VR*,*UTILIZZARE IL MARKET* ,*CHATBOT* .



State diagrams: Utilizzo piattaforma in funzione dell'utente .

GESTIONE PIATTAFORMA (ADMIN)

L' admin effettua l'accesso , come descritto in precedenza gestisce la piattaforma , gestendo il market , admin, ed eliminando utenti



5.2 System Architecture

Per quanto riguarda innanzitutto la definizione dell'architettura, quest'ultima come già anticipato sarà un'architettura di tipo **client-server**, con la possibilità di diverse connessioni client in contemporanea.

L'infrastruttura di rete che si mira ad utilizzare, sarà la rete **Internet**.

--Il ruolo centrale del server sarà quello di creare diversi thread per ogni connessione, in modo da poter gestire più richieste in contemporanea, di regolare l'accesso e la registrazione di ogni diverso utente, e di effettuare i calcoli necessari all'elaborazione del tipo di obiettivo in maniera completamente trasparente all'utente, non solo per quanto riguarda il tipo di algoritmo utilizzato, ma anche dal punto di vista dell'efficienza, che verrà così astratta dal tipo di computer utilizzato dal client.

Ciò permetterà di avere diversi vantaggi e qualche svantaggio:

PRO

- Possibilità di modificare, aggiungere o rimuovere differenti algoritmi di scheduling senza dover aggiornare il software lato client.
- Astrazione del tempo di calcolo richiesto rispetto al tipo di dispositivi client, in

questo modo anche su dispositivi meno performanti il calcolo verrà effettuato nello stesso quantitativo di tempo, la differenza dipenderà solamente dalla velocità della connessione.

- Possibilità di visualizzare e catalogare eventuali statistiche relative all'utilizzo del software, da poter utilizzare a fini di analisi o commerciali.
- Possibilità per l'utente di accedere al proprio diario alimentare in ogni momento e da diversi dispositivi.
- Possibilità di sviluppare diverse versioni del software funzionanti su altri tipi di dispositivi, quali tablet e smartphone, dovendo agire solo sul lato client.

CONTRO

- Necessità di avere una connessione alla rete per utilizzare il software.
- Possibilità di disservizi nel caso di guasti al server o di congestione nella rete.
- Overhead del tempo richiesto, dovuto alla trasmissione e alla ricezione dei dati.

Dopo aver valutato le caratteristiche positive e negative di tale scelta, si è giunti con il alla conclusione che l'utilizzo di un tipo di architettura distribuita possa

fornire un valore aggiunto al software, e si è dunque confermato l'avvio dello

sviluppo con questa metodologia. Questa architettura da sviluppare sarà divisa in due parti, la parte del server e la parte del client che dovranno interagire tra di loro come vedremo in seguito.

5.3Server

Il server sarà strutturato come un normalissimo server web: attenderà una richiesta di connessione da parte di un client sulla propria socket adibita al collegamento (che utilizzerà come porta di default la 4240) e, all'arrivo di una nuova richiesta di connessione, creerà una nuova socket di collegamento con il singolo client che verrà utilizzata per tutti i futuri messaggi.

Perché un'applicazione realizzata in [PHP](#) possa utilizzare le informazioni contenute all'interno di un database questa deve poter avere accesso ad esse, a questo scopo l'applicazione dovrà poter comunicare con l'RDBMS che gestisce la base di dati, ciò è possibile attraverso un procedura iniziale e necessaria chiamata "connessione"; per evitare ambiguità è bene chiarire che la procedura di connessione avviene tra lo script e il programma che gestisce la base di dati e non tra lo script e la base di dati stessa; una volta terminata la procedura necessaria per la connessione all'RDBMS, sarà possibile avviarne una seconda chiamata di "selezione" del database da utilizzare.

Per aprire una connessione da un'applicazione in PHP al database manager MySQL, si utilizza una funzione nativa del linguaggio chiamata `mysql_connect()`, essa restituisce un identificativo di connessione MySQL in caso di successo, diversamente restituisce *FALSE*;

questa funzione richiede il passaggio di tre parametri che sono argomenti della funzione(descritti in dettaglio successivamente):

hostname;username: password:

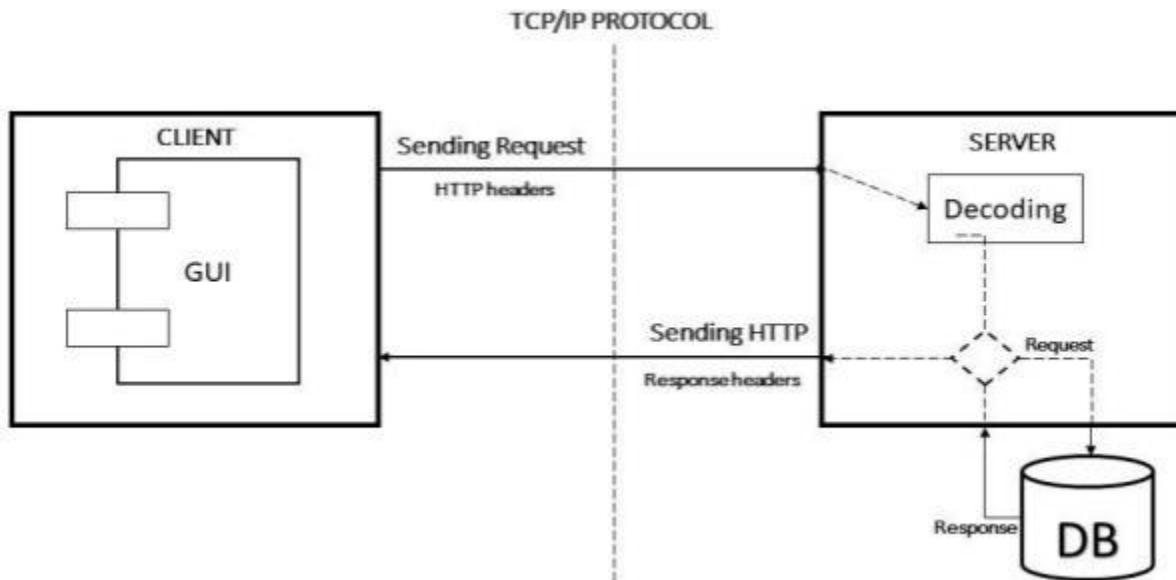


Figura 13 Architettura Client-Server

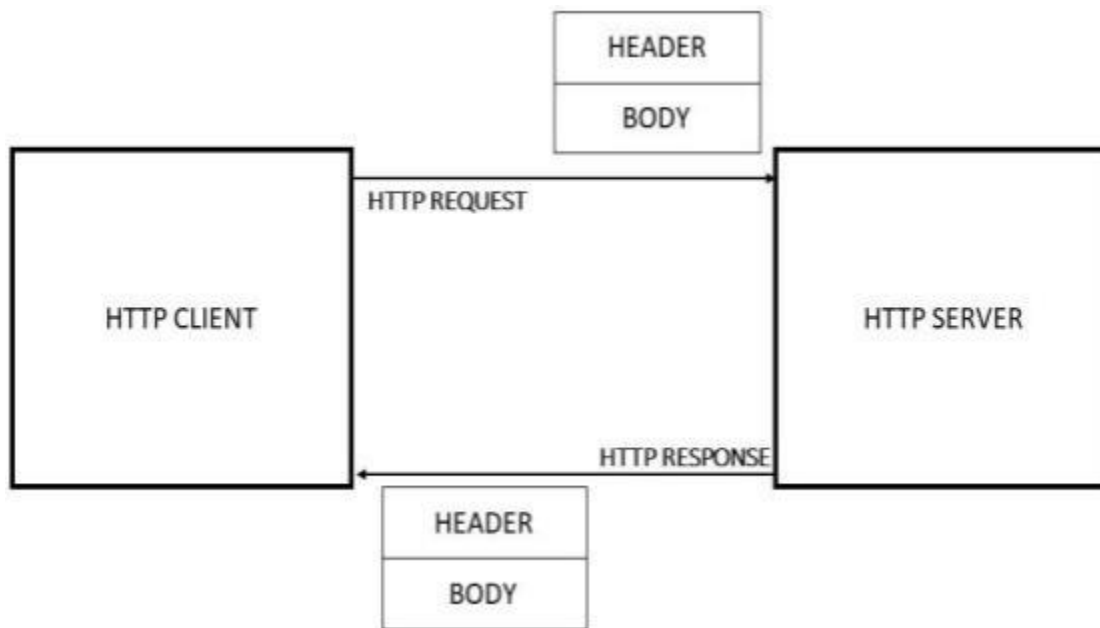
Communication Protocol http

Per quanto riguarda il protocollo da utilizzare per la comunicazione tra client e server, si è scelto l'impiego del protocollo http, che è il protocollo che sta alla base delle architetture software basate su tale metodologia.

Il protocollo prevede che il Client esegue una richiesta e il server restituisce la risposta, dunque a queste due operazioni susseguono numerosi scambi di messaggi tra i due componenti.

Per via della natura intrinseca del problema, l'unico tipo di protocollo di comunicazione utilizzabile è un protocollo sincrono, nel quale il client attende la risposta del server alla propria richiesta. Per questo motivo ogni messaggio inviato è di tipo bloccante per l'utente, che si ritroverà dunque ad attendere la relativa risposta da parte del server (fatta eccezione per il messaggio di chiusura connessione, il quale verrà inviato subito prima di effettuare il logout, in modo che il Client non rimanga bloccato nel caso di disservizi alla rete).

Per quanto riguarda il protocollo di trasporto, essendo fondamentale l'arrivo a destinazione dei dati così come questi sono stati inviati. Verrà utilizzato il protocollo TCP, il protocollo sicuro di trasporto orientato alla connessione



Protocollo HTTP (Request/Response)

5.4 Activity Diagram (diagramma attività)

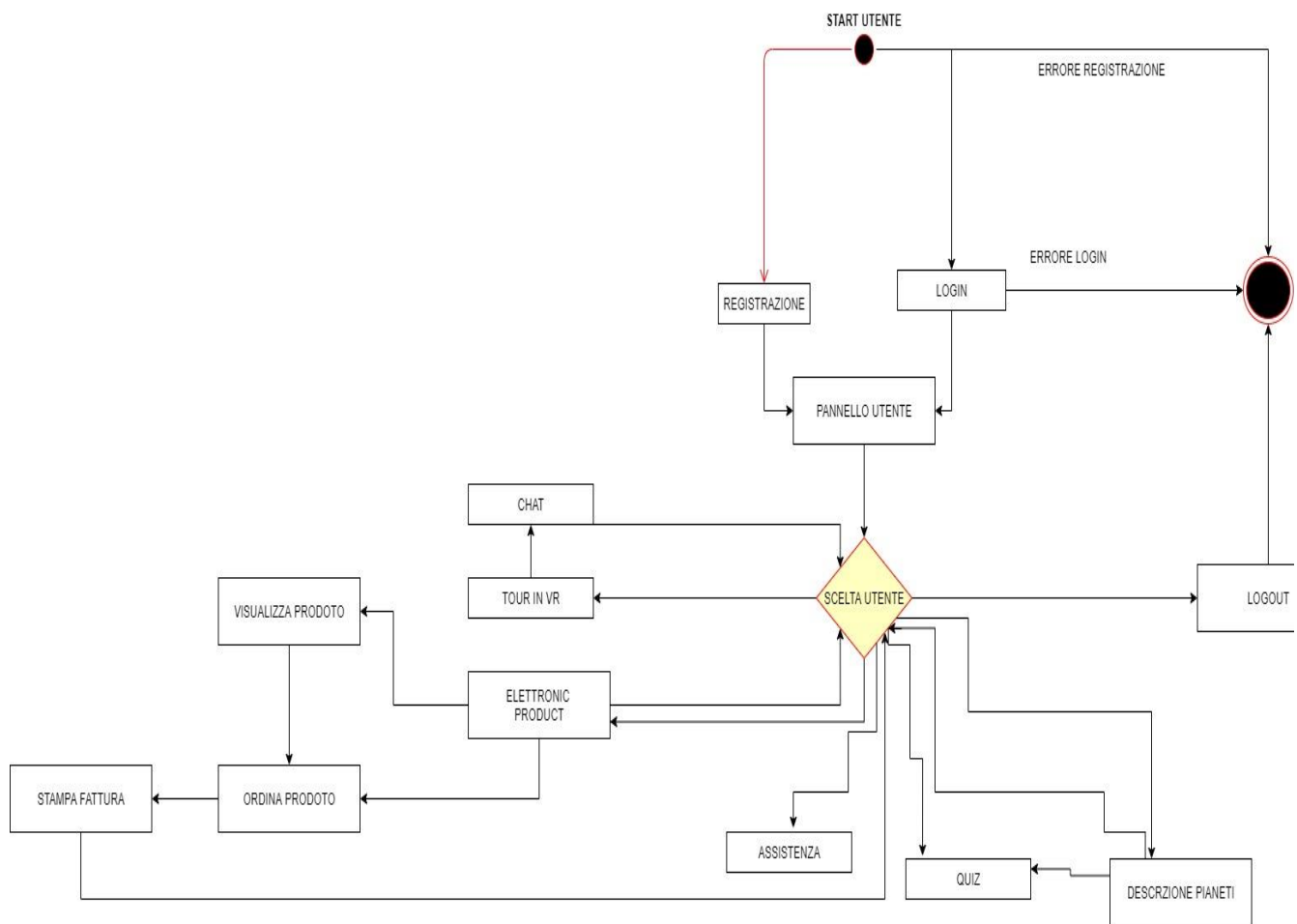
Avendo definito i requisiti dell'applicazione e schematizzato i vari casi d'uso di un'utente all'interno dell'applicazione, definendo anche scenari “straordinari”, adesso è possibile modellare i vari casi d'uso anche dal punto di vista grafico mediante gli activity diagram.

Questo diagramma si aggiunge alla documentazione che può essere presentata. La sua funzione è quella di prendere le descrizioni dei flussi sia quello principale che quelli alternativi, sono fusi nello stesso diagramma per poter far vedere l'insieme delle azioni compiute dal caso d'uso preso in considerazione.

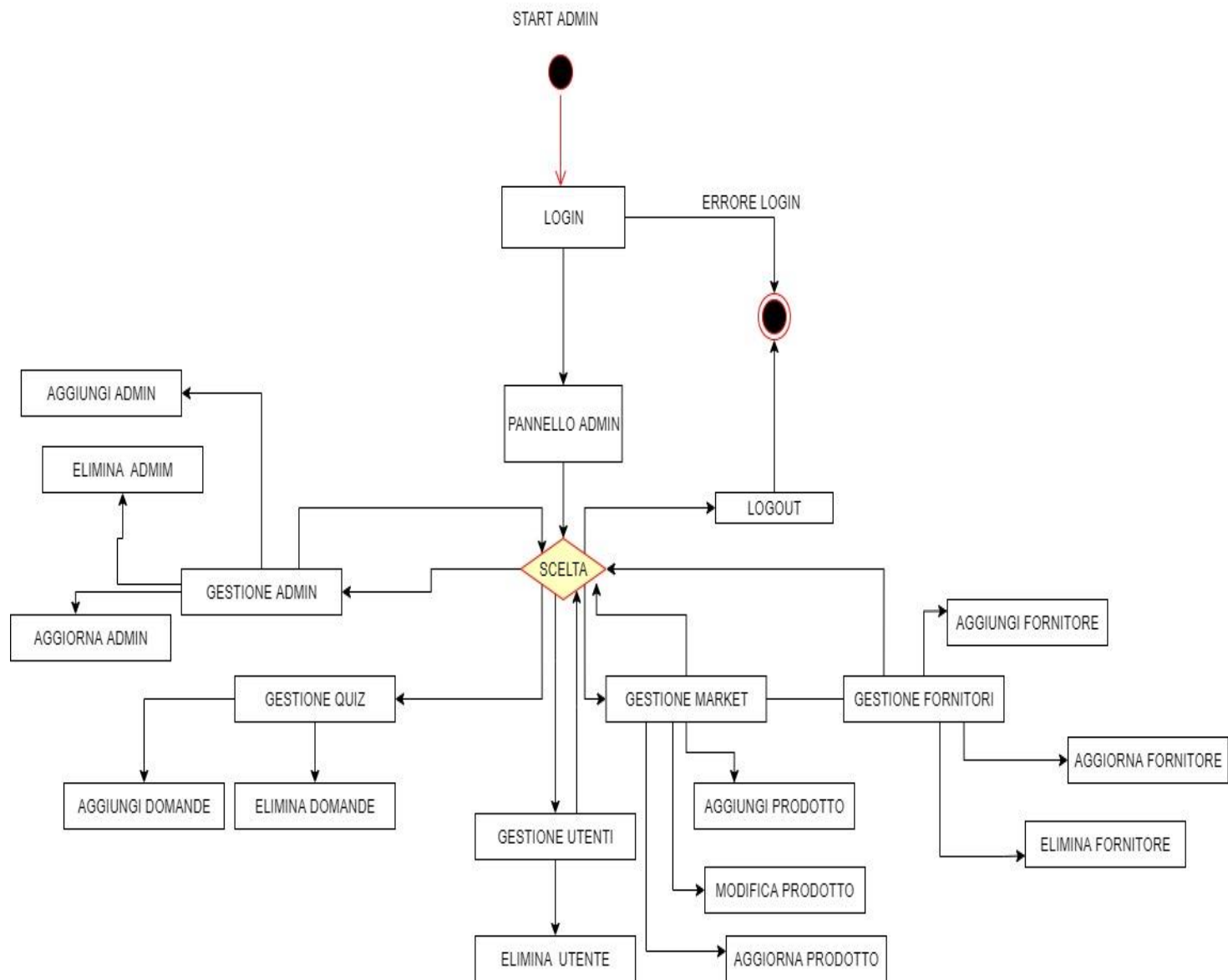
Legenda:

Simbolo	Significato
Cerchio pieno	Punto d'inizio.
Rettangolo con bordi smussati	Attività
Rombo	Punto di scelta

AD: *UTENTE*



AD: ADMIN



5.5 Descrizione delle GUI.

In questo paragrafo andremo a descrivere le interfacce che sono state realizzate per far interagire gli stakeholders con il nostro sistema. Per la realizzazione delle interfacce si è andati alla ricerca della semplicità e dell'usabilità. La struttura delle GUI si basa su un insieme di frame, al cui interno, tramite dei "button" ci sposteremo tra di essi. La struttura è abbastanza semplice in quanto ogni frame viene richiamato solo quando un evento voluto dall'utente lo farà comparire, gli eventi scaturiscono nel momento in cui viene cliccato un button.

Per la realizzazione delle GUI sono stati utilizzati framework come BOOTSTRAP/CSS/HTML/A-FRAME.

Per una maggiore comprensione delle interfacce le andremo a descrivere ognuna di esse singolarmente.

Nota: nelle barra di navigazione sono presenti i tasti per accedere ai social

REGISTRAZIONE UTENTE

REGISTRAZIONE:

Un utente per poter utilizzare l' applicazione, deve essere registrato al sistema. Per potersi registrare, un utente deve per prima cosa deve trovarsi nella sezione relativa

alla registrazione. Dopodiché compilare il modulo di registrazione inserendo un nome utente, un'email e una password, tutte queste informazioni saranno utili nella fase di login. Una volta compilato il modulo, per completare questa fase l'utente dovrà premere il relativo pulsante e che potrà dare due conclusioni: le informazioni inserite dall'utente sono corrette e quindi l'utente viene registrato al sistema, o le informazioni non sono corrette e quindi il sistema mostrerà il relativo messaggio di errore e l'utente potrà riprovare la registrazione correggendo gli eventuali errori.

The image shows a registration form for a service called "VR FOR ALL". The form is set against a dark red background. It includes several input fields with icons indicating the type of information required: a person icon for username, an email icon for email, a house icon for address, a phone icon for phone number, and a person icon for a second username or name. The fields are filled with the following text: "alessandro", "mad", "occ", "ale@gmail.com", "Via Cav.F. Sangiovanni 4", "871452555555", and "ale". A large orange button labeled "Create account" is positioned at the bottom of the form. In the top right corner, there are social media icons for Twitter, Facebook, and Instagram. The text "VR FOR ALL" is visible in the top left corner. A small copyright notice "© 2021 Designed and Coded by Furqan" is located at the bottom right of the form area.

VR FOR ALL

Twitter Facebook Instagram

@alessandro

@mad

@occ

ale@gmail.com

Via Cav.F. Sangiovanni 4

871452555555

ale

—

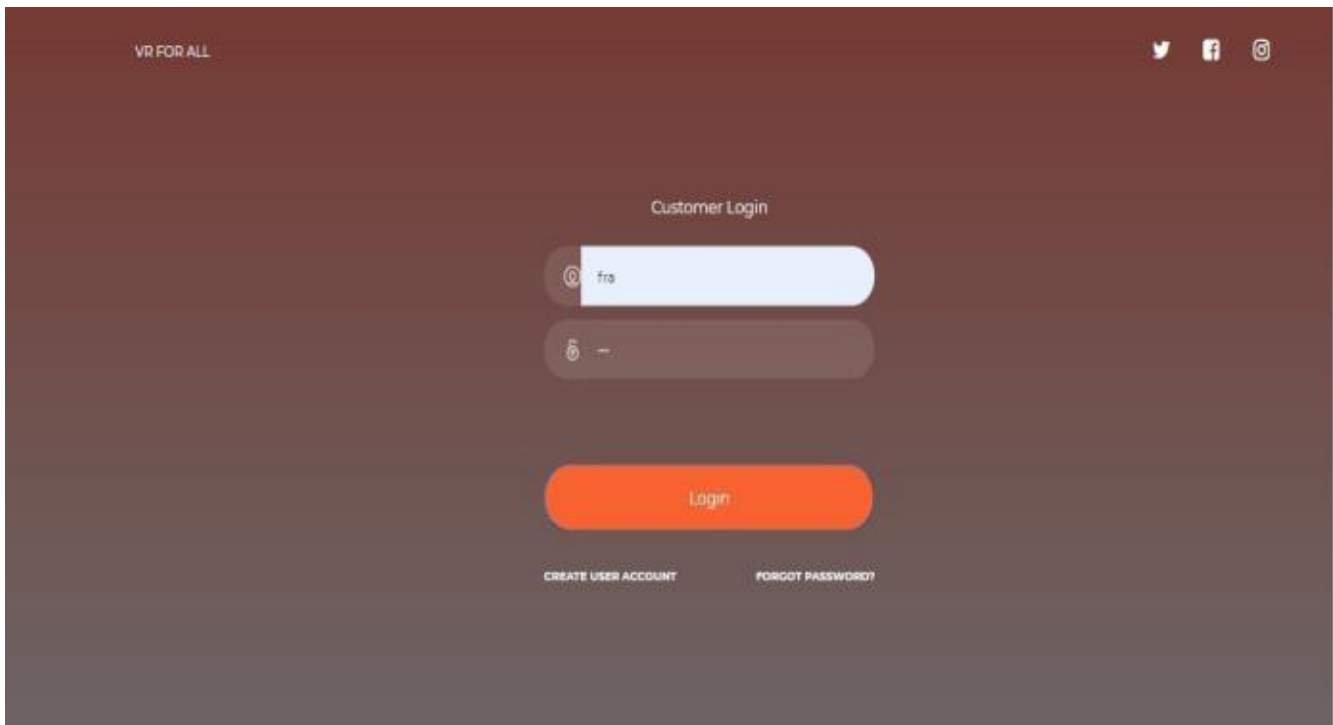
Create account

© 2021 Designed and Coded by Furqan

LOGIN UTENTE

LOGIN(UTENTE)

Dopo che un utente ha effettuato la registrazione, per poter usare la WEBVR, l'utente effettua il login. Nella fase di login l'utente dovrà compilare un modulo composto da un nome utente e da una password, che dovranno essere compilate con password e nome utente usate nella fase di registrazione. Dopo aver compilato il modulo e premuto il relativo pulsante il sistema controllerà che le informazioni inserite siano corrette e che appartengano ad un'utente registrato e se questo controllo va a buon fine l'utente sarà loggato al sistema e potrà usare il software , mentre se il controllo fallisce il sistema mostrerà un messaggio di errore e l'utente potrà ripeter e il login



The screenshot displays a login interface for a system named "VR FOR ALL". At the top left, the text "VR FOR ALL" is visible. At the top right, there are three social media icons: Twitter, Facebook, and Instagram. The main heading for the form is "Customer Login". Below this, there are two input fields: the first is for the username, containing the text "frs", and the second is for the password, which is currently empty. Both fields have a circular icon on the left side. Below the input fields is a large orange button labeled "Login". At the bottom of the form, there are two links: "CREATE USER ACCOUNT" on the left and "FORGOT PASSWORD?" on the right.

DASHBOARD(UTENTE)

La dashboard del software dovrà essere semplice e chiara. L'idea di base è quella di 5 bottoni, disposti orizzontalmente, attraverso i quali, all'utente, sarà possibile svolgere le operazioni principali del programma, l'utente dunque entrerà nelle aree dedicate (descritte successivamente)

ELECTRONIC PRODUCTS:

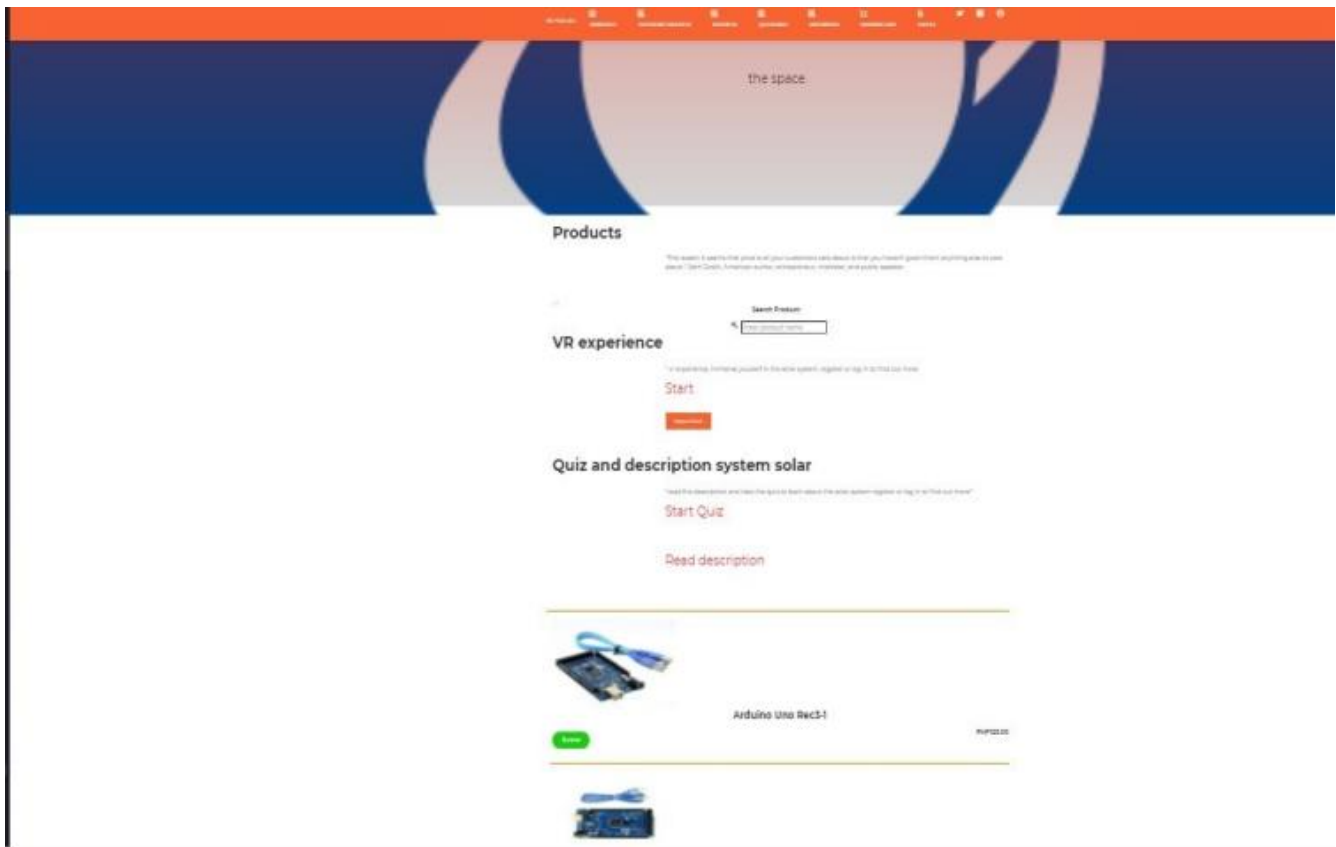
.TOUR IN VR:

.QUIZ PLANET :

.DESCRIPTION: .

.LOGOUT

Nella home page dell'area utente , è possibile accedere direttamente ad ognuno dei servizi descritti in precedenza , è possibile visitare il market e cercare direttamente il prodotto desiderato, o acquistarlo.



ELECTRONICS PRODUCT

ELECTRONIC PRODUCTS: *consentirà all'utente di visualizzare ed eventualmente acquistare un prodotto o aggiungerlo nel carrello tramite l'apposito pulsante(come mostrato in figura), acquistato il prodotto ,l'utente può stampare la fattura*

All Product List

Serial	Product Name	Description	Price(Php)	Quantity	Category	Option
11111	Arduio Uno Rev3-1	Small Arduio Uno Blue	125.00	0	Arduio	View
34156780	Arduio Mega	ATMega Arduio	155.00	2147483647	Arduio	View
33333333333333333333333333333333	clao	clao	33.00	333	Arduio	View
	456523702	Flame Sensor 3 Pins	455.00	444	Sensor	View
	45422791	Raspberry Pi 3 Model B+	760.00	221	Raspberry Pi	View
890	Sensor	Able to sense product	1500.00	679	Sensor	View
1353	X9 THOR - Gaming Mouse	7D Macro Programmable Gaming Mouse, Sensor: A714 Instant, LED: RGB 16.8 million colors, Interface : USB, DPI: 4800dpi, Cable Length 1.8m nylon braided, Supported OS: Windows Vista, Win7/8/10, Mac OS X 10.5 or later, Linux, Chrome OS	2200.00	17	Banana Pi	View



Serial number: 45422791
Product name: Raspberry Pi 3
Description: Model B+
Type: Raspberry Pi
Price: Php760.00
Items in stock: 221

Add to Cart

Francesco's Checking Out!

***** Your tracking number: 2010 | Total: Php760.00 | Tax: Php91.2 | Shipping Address: ffff Manila City *****

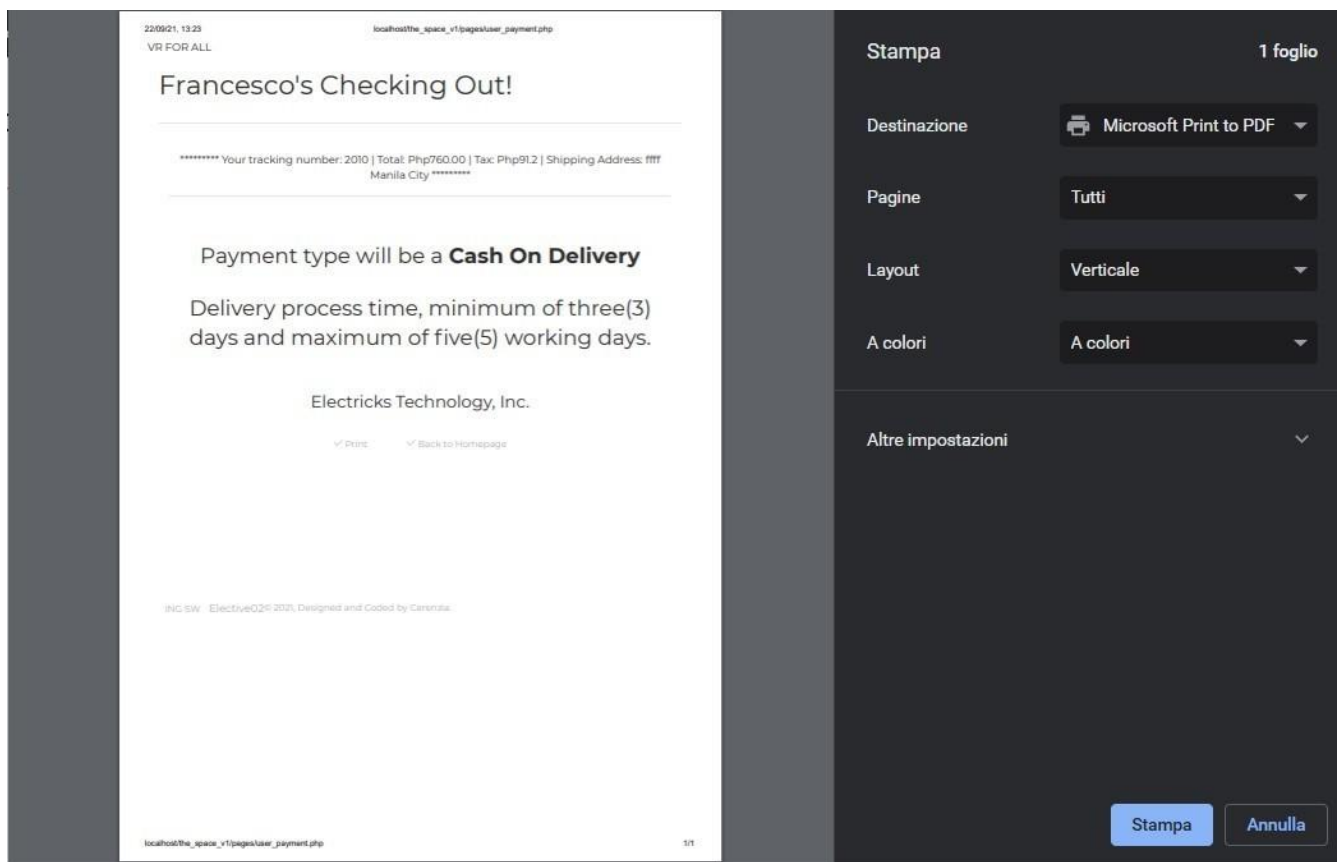
Payment type will be a **Cash On Delivery**

Delivery process time, minimum of three(3) days and maximum of five(5) working days.

Electricks Technology, Inc.

✓ Print

✓ Back to Homepage



funzionalità stampa

TOUR IN VR

TOUR IN VR: l'utente potrà immergersi in una realtà virtuale, dove visiterà il sistema solare, potrà navigare tra i pianeti con una voce che indicherà all'utente su quale pianeta sta andando, tutto il viaggio sarà accompagnato da una musica rilassante per un'esperienza ancora più immersiva. All'utente viene data la possibilità di commentare l'esperienza in una chat dedicata.

Nelle immagini sottostanti è riportato come si presenta il tour all'utente

CONTROLLO: *l'utente può controllare la navicella e la navigazione tramite la testa, dando la sensazione di gestire il tutto, per un'esperienza più immersiva, tramite il pallino che si vede in figura potrà scegliere la destinazione voluta. Inoltre l'utente potrà commentare l'esperienza in una chat dedicata,*



Image iniziale del tour

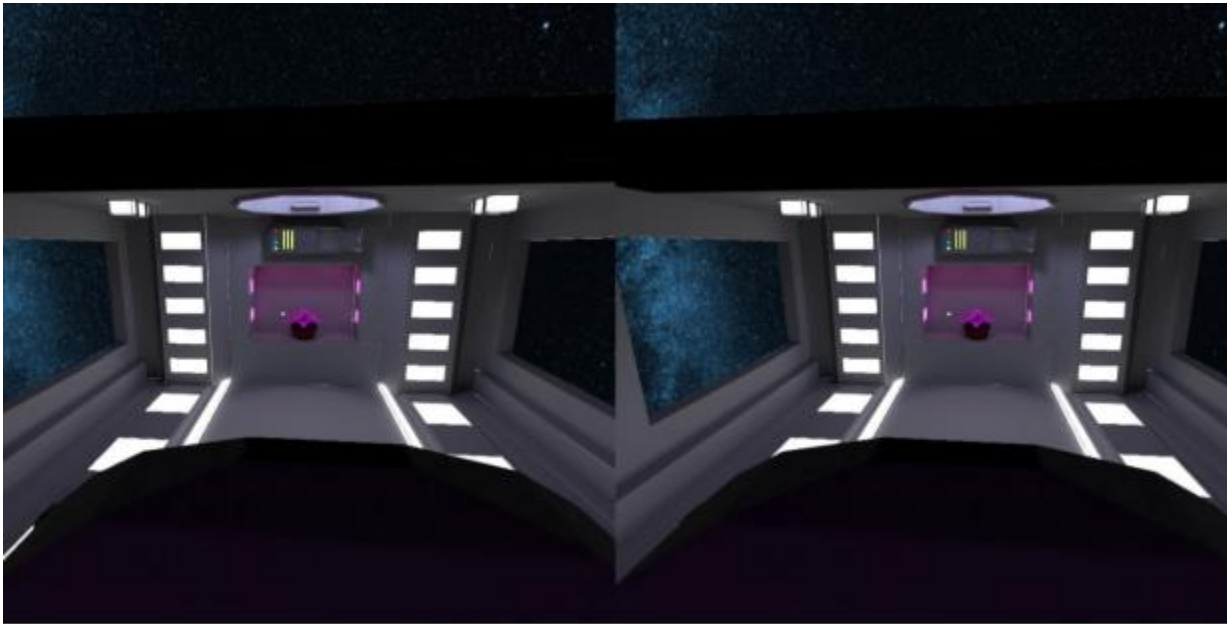
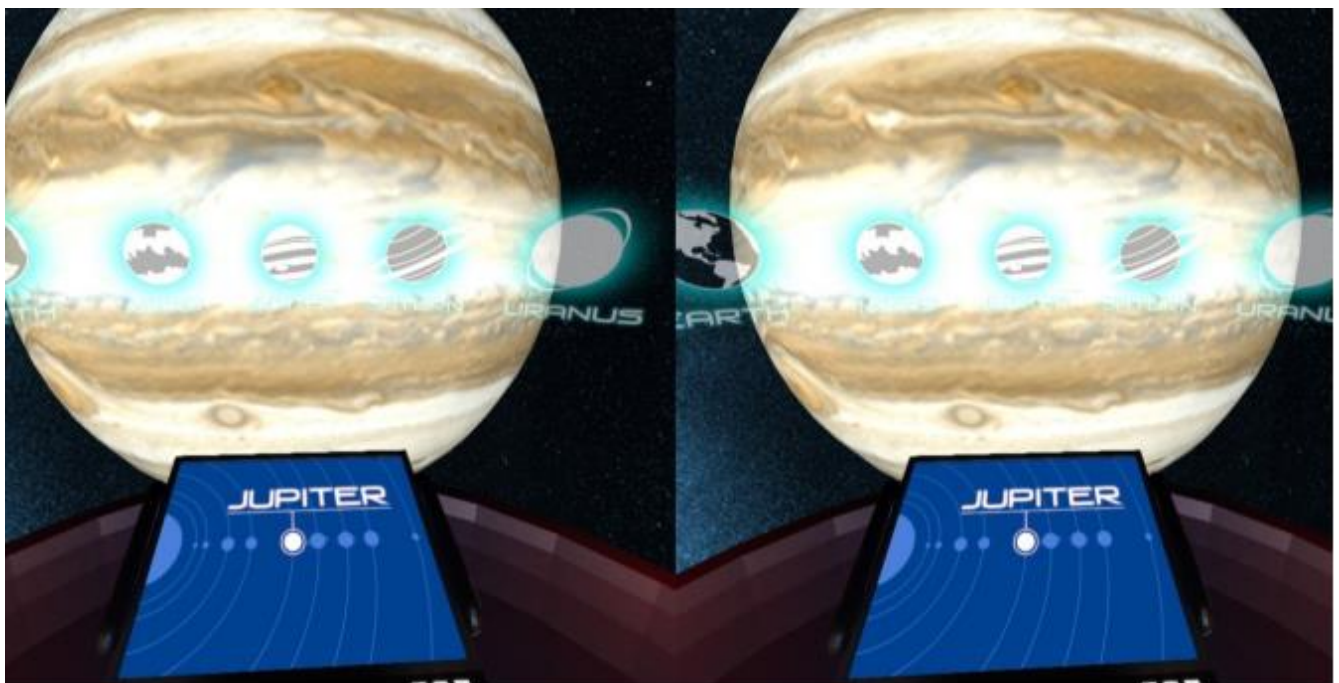


Immagine navicella

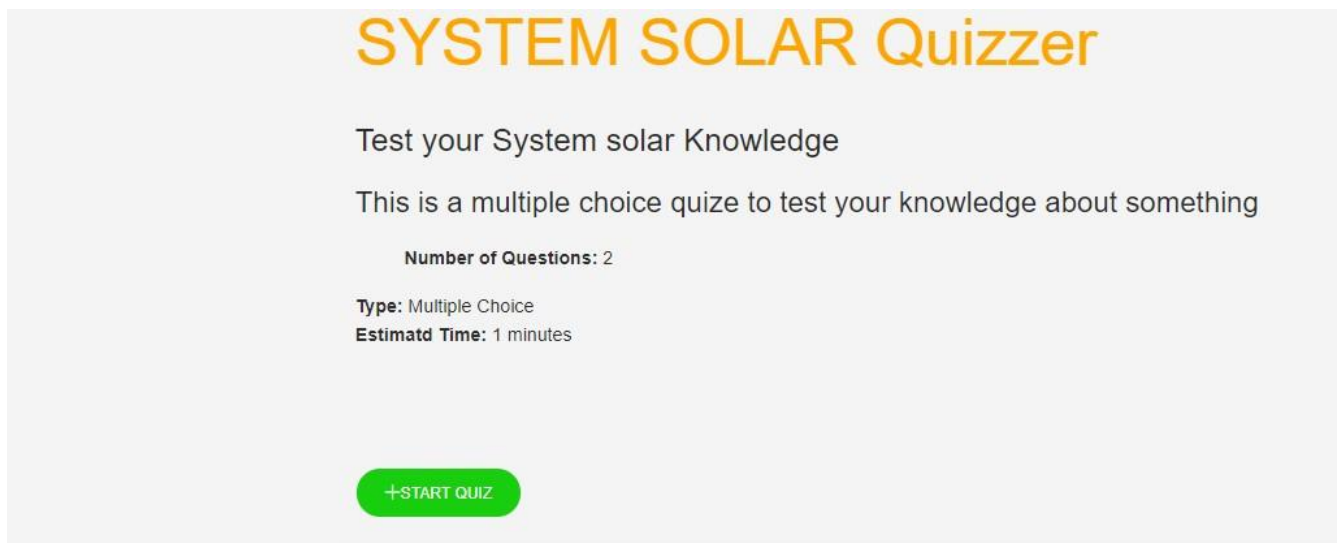


Esempio di come si presentano i pianeti

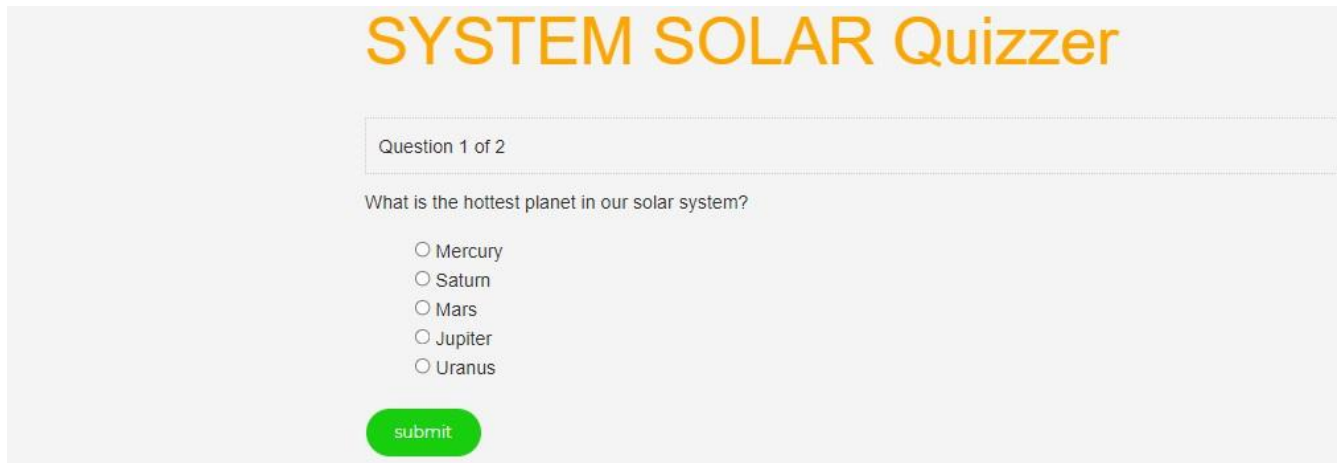


QUIZ PLANET

QUIZ PLANET : quiz basato sul sistema solare, sono domande a risposta multipla , a fine quiz l'utente vedrà le risposte esatte e ne verificherà la correttezza .mostrando alla fine del test la percentuale di risposte esatte, l'utente deve rispondere a tutte le domande , se ne salta 1 il sistema lo avviserà.



Schermata iniziale



SYSTEM SOLAR Quizzer

Question 1 of 2

What is the hottest planet in our solar system?

- ☐ Mercury
- ☐ Saturn
- ☐ Mars
- ☐ Jupiter
- ☐ Uranus

submit

esempio su come si presentano le domande



SYSTEM SOLAR Quizzer

You are Done!

Congrats! You have completed the test

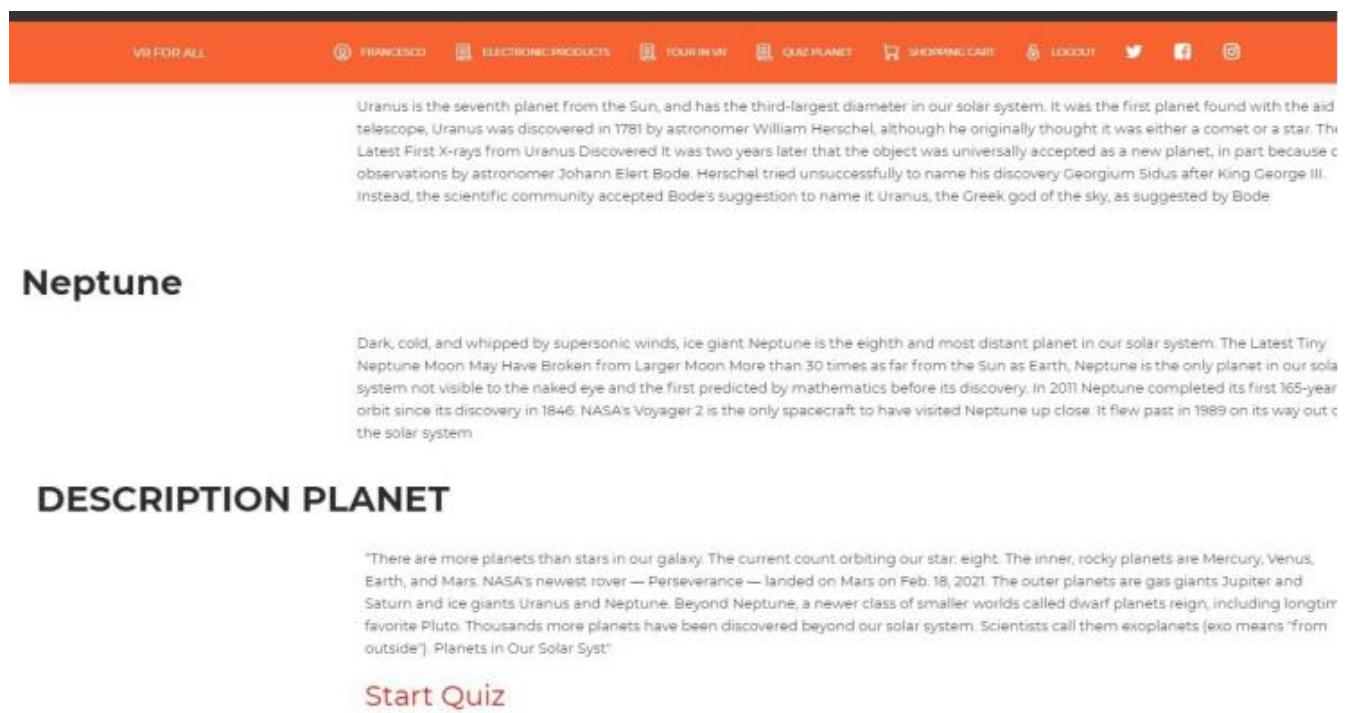
Final score: 0

+HOME +REPEAT QUIZ

Screen risultato test

DESCRIPTION

2. **DESCRIPTION:** *permette all'utente di leggere un'accurata descrizione del sistema solare . A fine pagine l'utente può tentare il quiz per verificare le conoscenze acquisite nella descrizione letta*



Earth

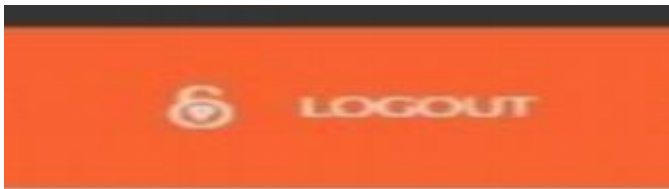
Our home planet is the third planet from the Sun, and the only place we know of so far that's inhabited by living things. The Latest Ocean Worlds Resources CUPID CubeSat Will Get New Perspective on Sun-Earth Boundary While Earth is only the fifth largest planet in the solar system, it is the only world in our solar system with liquid water on the surface. Just slightly larger than nearby Venus, Earth is the biggest of the four planets closest to the Sun, all of which are made of rock and metal. The name Earth is at least 1,000 years old. All of the planets, except for Earth, were named after Greek and Roman gods and goddesses. However, the name Earth is a Germanic word, which simply means "the ground."

Mars

Mars is the fourth planet from the Sun – a dusty, cold, desert world with a very thin atmosphere. Mars is also a dynamic planet with seasons, polar ice caps, canyons, extinct volcanoes, and evidence that it was even more active in the past. The Latest from Mars NASA's Perseverance Rover Collects Puzzle Pieces of Mars' History Mars is one of the most explored bodies in our solar system, and it's the only planet where we've sent rovers to roam the alien landscape. NASA currently has two rovers (Curiosity and Perseverance), one lander (InSight), and one helicopter (Ingenuity) exploring the surface of Mars. Perseverance rover – the largest, most advanced rover NASA has sent to another world – touched down on Mars on Feb. 18, 2021, after a 203-day journey traversing 293 million miles (472 million kilometers). The Ingenuity helicopter rode to Mars attached to the belly of Perseverance. Perseverance is one of three spacecraft that arrived at Mars in 2021. The Hope orbiter from the United Arab Emirates arrived on Feb. 9, 2021. China's Tianwen-1 mission arrived on Feb. 10, 2021, and includes an orbiter, a lander, and a rover. Europa and India also have spacecraft studying Mars from orbit. In May 2021, China became the second nation to ever land successfully on Mars when its Zhurong Mars rover touched down. An international fleet of eight orbiters is studying the Red Planet from above including three NASA orbiters: 2001 Mars Odyssey, Mars Reconnaissance Orbiter, and MAVEN. These robotic explorers have found lots of evidence that Mars was much wetter and warmer, with a thicker atmosphere, billions of years ago.

LOGOUT

LOGUT: *termina la sessione, l'utente verrà reindirizzato nella home page generale del software*



Bottone per effettuare il logout

CHATBOT:

L'utente tramite un area dedicata può utilizzare l'assistenza a lui dedicata.

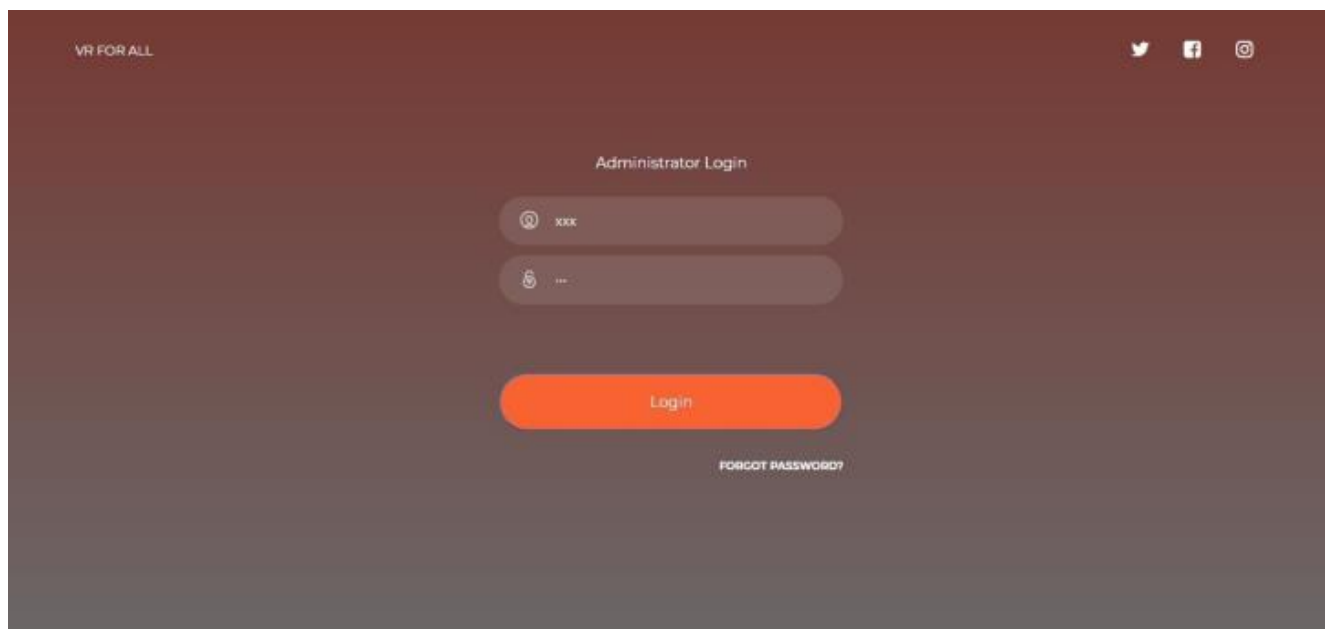
In questa sezione l'utente che utilizza la piattaforma può decidere se far mostrare al bot informazione sui pianeti , o sulla realtà virtuale, ma può anche accedere alle funzionalità del software, garantendo all'utente un' assistenza H24 . questa funzionalità permette un'utilizzo più semplificato della piattaforma.



LOGIN ADMIN

LOGIN(ADMIN)

Dopo che un utente ha effettuato la registrazione, per poter usare la WEBVR, l'utente effettua il login. Nella fase di login l'utente dovrà compilare un modulo composto da un nome utente e da una password, che dovranno essere compilate con password e nome utente usate nella fase di registrazione. Dopo aver compilato il modulo e premuto il relativo pulsante il sistema controllerà che le informazioni inserite siano corrette e che appartengano ad un'utente registrato e se questo controllo va a buon fine l'utente sarà loggato al sistema e potrà usare il software , mentre se il controllo fallisce il sistema mostrerà un messaggio di errore e l'utente potrà ripetere il login.



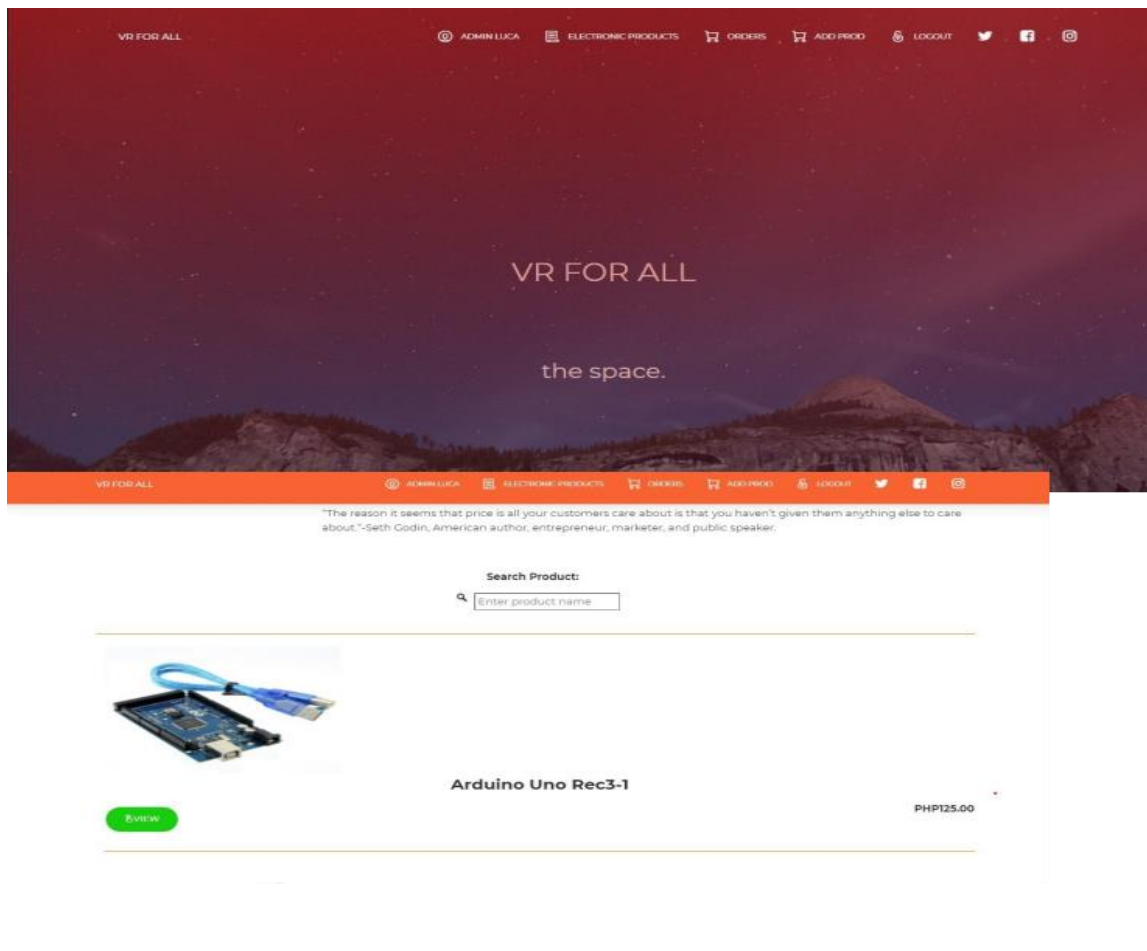
The image shows a web interface for "VR FOR ALL" with a dark red background. In the top left corner, the text "VR FOR ALL" is visible. In the top right corner, there are three social media icons: Twitter, Facebook, and Instagram. The main heading is "Administrator Login". Below this, there are two input fields: the first is for the username, labeled with a user icon and the text "XXXX", and the second is for the password, labeled with a lock icon and the text "XXXX". Below these fields is a large orange button labeled "Login". At the bottom, there is a link that says "FORGOT PASSWORD?".

DASHBOARD(ADMIN)

La dashboard del software dovrà essere semplice e chiara. L'idea di base è quella di 3 bottoni, disposti orizzontalmente, attraverso i quali, all'utente, sarà possibile svolgere le operazioni principali del programma, cioè;eliminare un utente registrato,gestire ill market gestire gli admin del software.

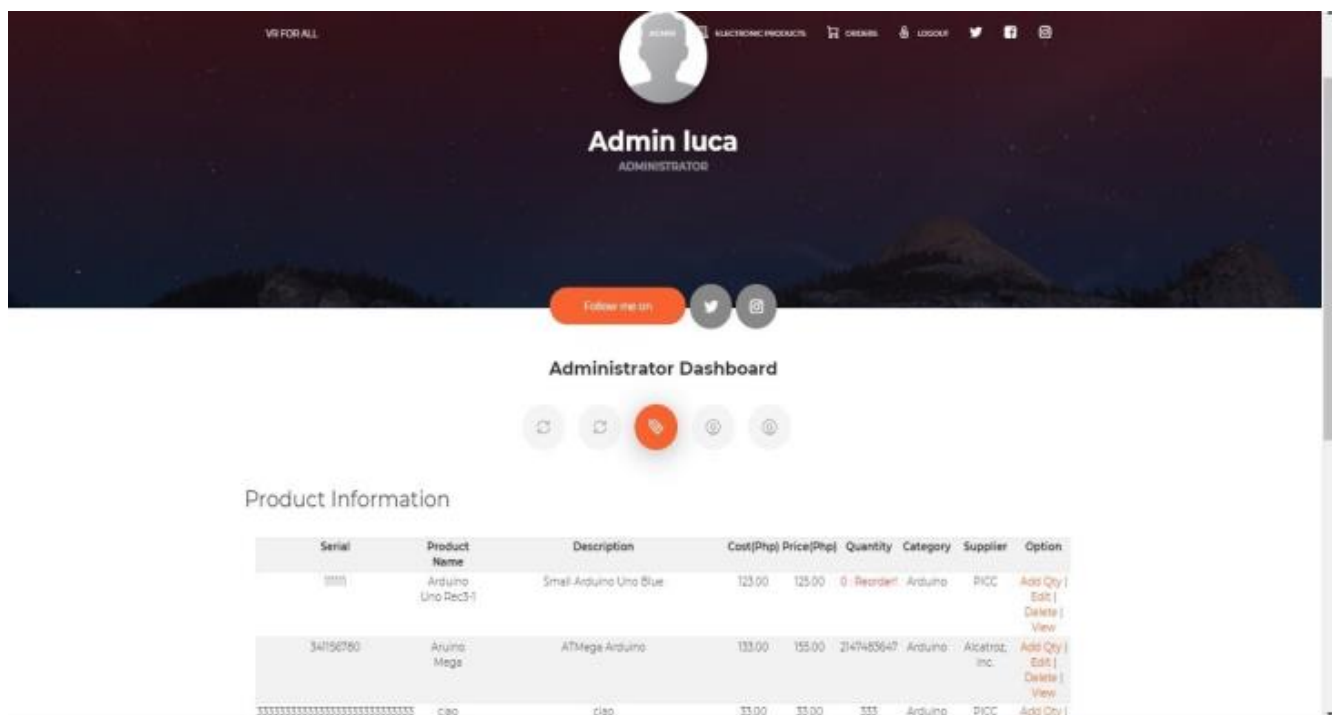
1.NOME ADMIN : 2.ELECTRONIC PRODUCTS : 3. ADD. PRODUCT:

Scorrendo nella pagina dedicata , sarà possibile visualizzare i prodotti o cercarne uno specifico



NOME ADMIN

NOME ADMIN : sarà data, a questa tipologia di utente la possibilità, di accedere ad una determinata area ,dove verranno mostrate le specifiche della piattaforma(CREDENZIALI UTENTE;ELIMINA UTENTE;OPERAZIONI SUL MARKET ; OPERAZIONI SUGLI ADMIN)



L'admin in questa are gestisce le parti coinvolte nella pittaforma

Purchased Product Information

[Back to Index](#)

Add Product Quantity

Product Name: Arduino Mega

Product Serial: 34756780

Name: Arduino Mega

Description: ATmega Arduino

Product Cost (Php): 133.00

Product Supplier: Arduino

Product Category: Alcatraz, Inc.

Available Quantity: 2147483647

Number of Quantity/Volume to be added:

[Add Quantity](#)

Purchased Product Information

[Back to Index](#)

Update Product

Serial number:

Product Name:

Product Description:

Product Cost (Php):

Product Price (Php):

Supplier:

Category:

Available Quantity: 2147483647 Pcs.

[Update Product](#)

In questa pagina , l'admin aggiunge la quantità di un prodotto sia nel caso fosse terminato o meno

In quest'area l'admin aggiorna il prodotto

ELECTRONIC PRODUCT

ELECTRONIC PRODUCTS : permette all'utente di visualizzare i prodotti contenuti all'interno del market ed effettuare eventuali modifiche .

All Product List

[← Back to index](#)

Serial	Product Name	Description	Price(Php)	Quantity	Category	Option
111111	Arduino Uno Rec3-1	Small Arduino Uno Blue	125.00	0	Arduino	View
341156780	Aruino Mega	ATMEGA Arduino	195.00	2147483647	Arduino	View
33333333333333333333333333333333	clao	clao	33.00	333	Arduino	View
456523702	Flame Sensor	Flame Sensor 3 Pins	455.00	444	Sensor	View
45422791	Raspberry Pi 3	Model B+	760.00	221	Raspberry Pi	View
890	Sensor	Able to sense product	1500.00	679	Sensor	View
1353	X9 THOR - Gaming Mouse	7D Macro Programmable Gaming Mouse, Sensor: A714 Instant, LED: RGB 16.8 million colors, Interface : USB, DPI: 4800dpi, Cable Length: 1.8m nylon braided, Supported OS: Windows Vista, Win7/8/10, Mac OS X 10.5 or later, Linux, Chrome OS	2200.00	17	Banana Pi	View

ADD PRODUCT

ADD. PRODUCT: *l'utente può inserire un nuovo prodotti all'interno del market con le relative informazioni, questa funzione serve a mantenere un market sempre*

Add Purchased Products

Product Name:

Product Name

Product Description:

Product Description

Product Cost (Php):

Value:eg. 123.4

Product Price (Php):

Value:eg. 123.4

Quantity:

Value:eg. 123

Supplier:

Amaz

Category:

Arduino

Picture 1 (Front View):

Scegli file Nessun file selezionato

Picture 2 (Side View):

Scegli file Nessun file selezionato

Picture 3 (Specifications):

Scegli file Nessun file selezionato

Serial:

Value:eg. 1234

✓ Add Product

aggiornato

AGGIUNTA FORNITORI:

l' admin può gestire i fornitori del market , aggiungendone di nuovi.

Supplier Information



Supplier Information

Supplier Name	Address	Contact	Email	Option
Amaz	Via Cav.F.Sangiovanni 4	000	amaz@gmail.com	Edit Delete
PICC	Manila, Phils.	(987)-884-12	picc@email.moto!	Edit Delete
QUEZELCO	Infanta, Quezon	45643534567879	emal	Edit Delete

+ Add Supplier

Come si può vedere può anche eliminare o modificare quelli esistenti direttamente da questa pagina.

ADD QUESTION

ADD. QUESTION: *l'utente può inserire nuove domande all'interno del database con le relative scelte, questa funzione serve a mantenere il quiz sempre aggiornato.*

Da questa sezione si può andare ad eliminare anche le domande sui quiz.

SYSTEM SOLAR QUIZZER!!!

Add A question

Question Number

Question

Choice #1:

Choice #2:

Choice #3:

Choice #4:

Choice #5:

Correct choice number

DELETE A QUESTION

question_number	questions	Option
1	What is the hottest planet in our solar system?	Delete
2	What is the hottest planet in our solar system?	Delete

Implementation guidelines

Obiettivo: lo scopo della nostra piattaforma è: l'utente tramite l'esperienza in VR ,una descrizione , un quiz, acquisirà una conoscenza sul sistema solare , nel market sono presenti prodotti elettronici , compresi i visori per utilizzare al meglio la

piattaforma

Criterio: il criterio utilizzato si basa sull' apprensione dell'utente presso la piattaforma. L'utente tramite le varie funzionalità offerte, ha la possibilità di imparare divertendosi, inoltre può fare acquisir per le proprie passioni.

Presupposti:

- ❖ *Ogni utente ha una base di conoscenza differente sul sistema solare ;*
- ❖ *Ogni utente ha le proprie tempistiche;*
- ❖ *Ogni utente ha i propri interessi sull'elettronica;*
- ❖ *Ogni utente desidera il market aggiornato e l'assistenza sempre funzionante*

6Implementation Phase

In questa fase, successiva a quella di progettazione, si passa all'implementazione vera e propria.

6.1 Configuration DBMS (SERVER)

Per quanto riguarda la connessione tra client e server e la connessione tra il server e il database, si è reso necessario inserire diversi parametri di configurazione, quali ad esempio l'**indirizzo ip**, la **porta** del server e del database e il tipo di **dbms** utilizzato.

PHP inizializzerà la connessione con il DBMS come descritto in precedenza (4.2.2) tramite lo script sotto riportato.

```
$dbconn = mysqli_connect("localhost","root","","electricks");
```

```
// Check connection
```

```
if (mysqli_connect_errno())
```

```
{
```

```
echo "Failed to connect to MySQL: " . mysqli_connect_error();
```

}

?>

1. **hostname:** è il nome dell'host (o macchina ospitante) relativa al database manager MySQL a cui si desidera effettuare una connessione, esso identifica univocamente una postazione in Rete e può essere espresso sotto forma di indirizzo IP o stringa eventualmente seguita dal numero della porta attraverso cui l'RDMS attende le chiamate da parte dei client (i computer degli utenti che intendono interrogare i database), nel caso di un'installazione locale l'hostname è generalmente chiamato "localhost";
2. **username:** è il nome dell'utente abilitato alla connessione e alla manipolazione di uno o più database; MySQL prevede un utente iniziale che è quello di root a cui sono associati i privilegi per la manipolazione delle basi di dati gestite, l'utilizzatore potrà poi creare altri utenti a cui associare un username e privilegi comparabili o inferiori a quelli previsti per il root;
3. **password:** per questioni di sicurezza è buona norma associare una password ad ogni nuovo utente MySQL creato, questa permetterà di autenticarlo al momento della connessione con il Database manager

6.2Database

Per rispettare i requisiti forniti dal cliente e dunque poter gestire un grande numero di utenti, servirà utilizzare un database. Il database, utilizzando un approccio distribuito, sarà il fulcro per consentire la gestione di più utenti, e dunque anche la registrazione.

Inoltre, il database in questione terrà traccia degli ordini degli utenti in modo tale che queste informazioni, vengano rese accessibili da ogni device dal quale l'utente tenterà di utilizzare il software, tramite la dashboard avrà un resoconto generale del profilo . Per lo stesso principio anche la gestione della piattaforma deve essere gestita mediante database, in quanto un admin che accede da un diverso device deve poter avere la possibilità di accedere al database per gestire admin ,utenti e prodotti ,in modo tale da andare oltre al concetto di “locale” ed entrare in una logica “distribuita”per mezzo della rete internet

	Tabella	Azione	Righe	Tipo	Codifica caratteri	Dimensione	Overhead
<input type="checkbox"/>	admin	★ Mostra Struttura Cerca Inserisci Svuota Elimina	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	category	★ Mostra Struttura Cerca Inserisci Svuota Elimina	10	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	chat	★ Mostra Struttura Cerca Inserisci Svuota Elimina	7	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	choices	★ Mostra Struttura Cerca Inserisci Svuota Elimina	20	InnoDB	utf16_bin	16.0 KiB	-
<input type="checkbox"/>	customer	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	logs	★ Mostra Struttura Cerca Inserisci Svuota Elimina	192	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	order	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	order_details	★ Mostra Struttura Cerca Inserisci Svuota Elimina	60	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	payment	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	products	★ Mostra Struttura Cerca Inserisci Svuota Elimina	6	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	questions	★ Mostra Struttura Cerca Inserisci Svuota Elimina	2	InnoDB	utf16_bin	16.0 KiB	-
<input type="checkbox"/>	sales	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	sales_details	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	supplier	★ Mostra Struttura Cerca Inserisci Svuota Elimina	3	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	temp_trans	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	transactions	★ Mostra Struttura Cerca Inserisci Svuota Elimina	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
<input type="checkbox"/>	users	★ Mostra Struttura Cerca Inserisci Svuota Elimina	6	InnoDB	latin1_swedish_ci	16.0 KiB	-

Essendo in un'architettura client-server, l'utente dovrà interrogare il server per avere accesso alle informazioni del database.

Le operazioni effettuabili dal client del server sul database,sono :

- + OPERAZIONE DI LOGIN E REGISTRAZIONE DELL'UTENTE
- + OPERAZIONE DI PRELEVAMENTO E MEMORIZZAZIONE DATI UTENTI
- + OPERAZIONE DI PRELEVAMENTO E MEMORIZZAZIONE ORDIN
- + OPERAZIONE DI CHAT TRA UTENTI
- + OPERZIONE GESTIONE QUIZ

La scelta di utilizzare un sistema che gestisse il database non è casuale.

Un DBMS (DataBase Management System) permette di accedere in modo semplice e efficiente ad una base di dati mantenendone la consistenza, la privatezza e l'affidabilità. I vantaggi dell'uso di un DBMS sono molteplici:

- 1) *Affidabilità dei dati. Un DBMS offre dei metodi per salvare copie dei dati (backup) e per ripristinare lo stato della base di dati in caso di guasti software e hardware (recovery).*
- 2) *Accesso ai dati tramite un linguaggio universale. Ogni DBMS di una certa tipologia mette a disposizione un linguaggio di interrogazione (SQL nel caso relazionale). Tale linguaggio permette la creazione delle strutture che contengono i dati, l'inserimento, la cancellazione, l'aggiornamento dei dati e il recupero delle informazioni dalla base di dati.*
- 3) *Accesso concorrente ai dati. Un DBMS permetta a più utenti di accedere contemporaneamente alla base di dati. Più utenti possono accedere nello stesso istante a dati diversi. Inoltre, un DBMS fa in modo che l'accesso concorrente agli stessi dati non generi anomalie, cioè inconsistenza nello stato della base di dati rispetto alla realtà modellata.*

Quest'ultimo punto ha spinto l'intero team nella fase dell'implementazione di un'architettura client-server, che dai requisiti del cliente dovesse permettere ad un'utente di poter accedere da più dispositivi, di creare una base di dati modellata ad hoc. Tale scelta inoltre migliora l'organizzazione dal punto di vista strutturale e

ne migliora la velocità e la fluidità gestionale del software. Oltre i tecnicismi, va definito che in particolare l'uso di un database deve essere necessariamente di supporto, in quanto è un software le cui componenti sono fortemente agganciate, ad esso.

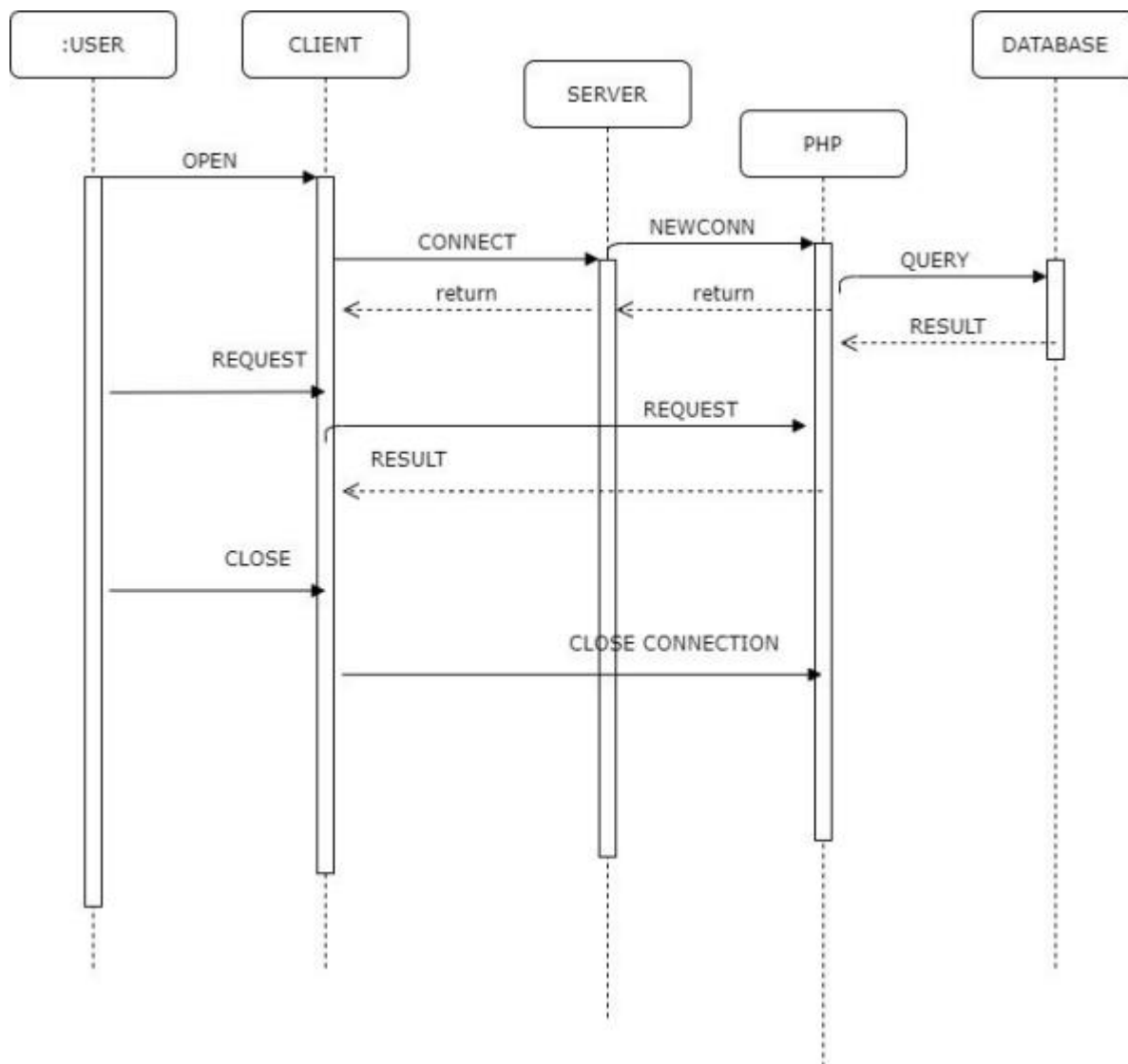
6.3 Client

Per quanto riguarda il client, quest'ultimo avrà una struttura semplice, in cui avremo esclusivamente le funzioni legate all'interazione con l'utente. Il client come ribadito più volte tramite le interfacce grafiche, effettueranno richieste al server. Sarà necessario aggiungere un'interfaccia iniziale per il login/la registrazione dell'utente, in quanto questa architettura viene scelta proprio per permettere l'accesso non solo multiplo tra più utenti, ma anche l'accesso da più dispositivi. Come descritto nei paragrafi precedenti, le interfacce sono semplici ed intuitive. Le funzionalità messe a disposizione degli utenti, sono molteplici, tutte pensate per garantire all'utente un corretto utilizzo della piattaforma. Una volta effettuata l'implementazione dell'architettura, il programma dovrà avere il seguente comportamento:

6.4 DIAGRAMMA DI SEQUENZA.

Come si potrà vedere dalle figure che verranno inserite in seguito tutte le operazioni fatte dagli stakeholder, quindi dal client verranno prima processate dal server e solo dopo inviate al database per poter essere eseguite, dopodiché il server invierà al client un messaggio di errore oppure di conferma delle operazioni, sarà quindi il client ad

occuparsi di mostrarlo all'utente. Questo viene fatto perché si è deciso di utilizzare l'architettura client-server descritta in precedenza



SEQUENCE DIAGRAM

DIAGRAMMA DI SEQUENZA UTENTE.

Per prima cosa il nostro utente deve registrarsi, se gli input inseriti durante la registrazione sono corretti il nostro nuovo utente verrà aggiunto al sistema. In seguito l'utente deve effettuare il log-in, se le credenziali saranno corrette l'utente ha accesso alla propria interfaccia.

inoltre il software per l'utilizzo di alcuni servizi utilizza il supporto al database (per un maggiore robustezza e correttezza). Queste funzionalità, come si può vedere nella figura sottostante, sono: il quiz sui pianeti, ordine prodotto, chat utenti

Per quanto riguarda il quiz, le risposte verranno inviate al server, tramite le opportune query, confronta la risposta corretta nel database, esito positivo, restituisce il risultato del test all'utente.

Ragionamento analogo vale per l'ordine di un prodotto. Se l'utente inserisce correttamente i campi (quantità, fornitori) verrà effettuato l'ordine, con la possibilità di stampare, altrimenti verrà visualizzato un messaggio di errore.

Differente il caso delle chat, l'utente inserisce il messaggio (feedback tour), verrà inserito nel database (se non lasciato vuoto), verrà visualizzato subito insieme ai messaggi di tutti gli utenti del database.

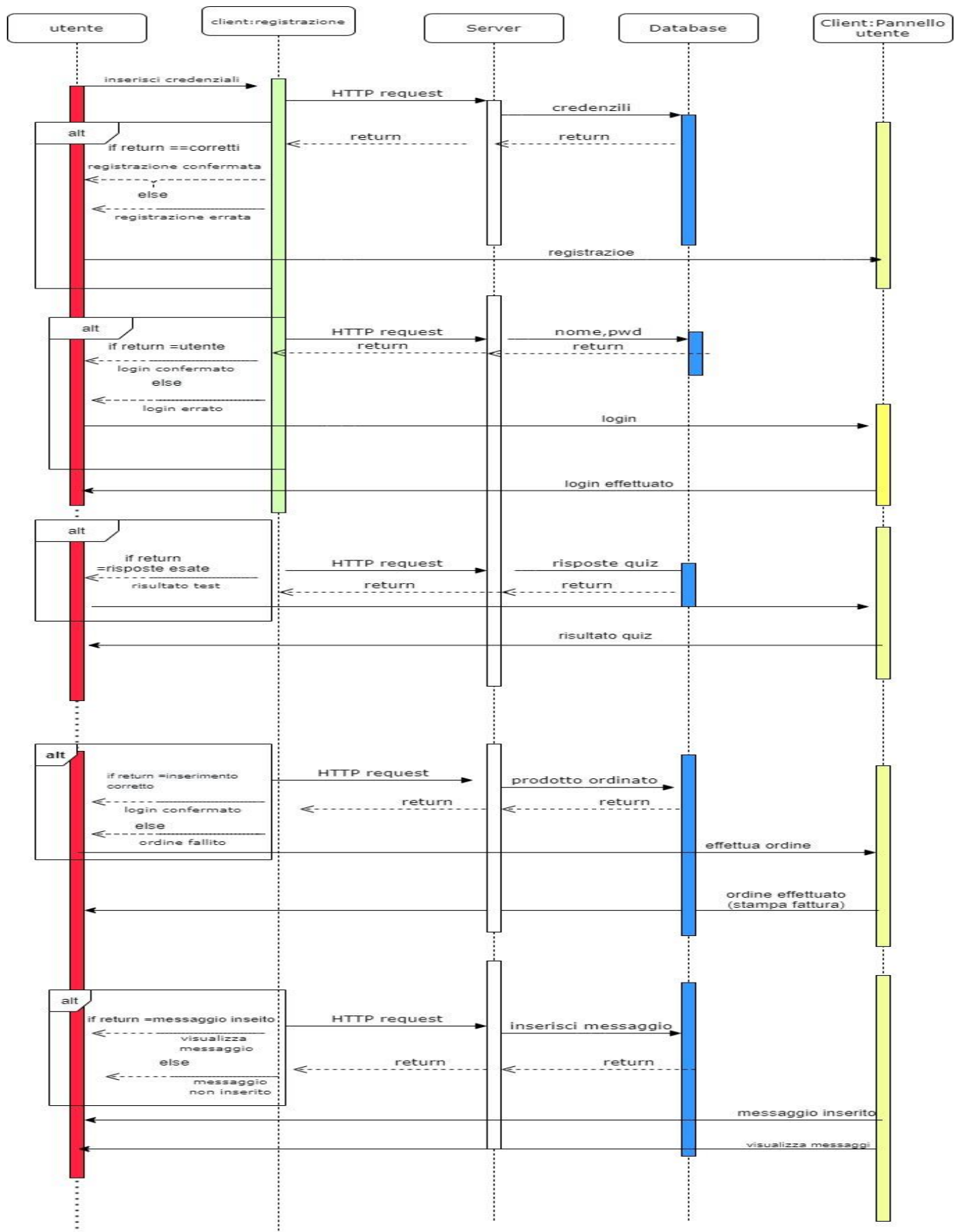


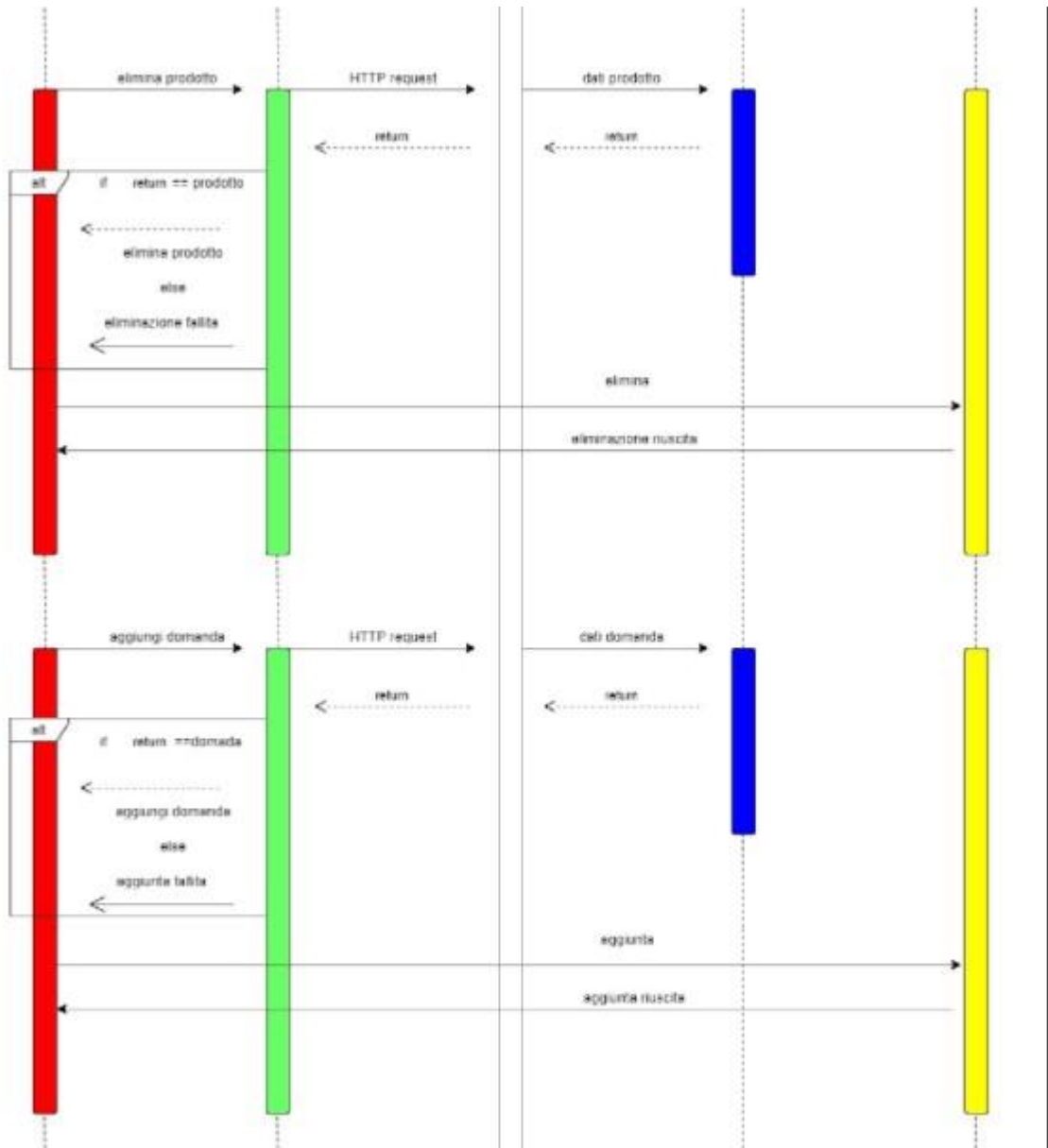
DIAGRAMMA SEQUENZA ADMIN

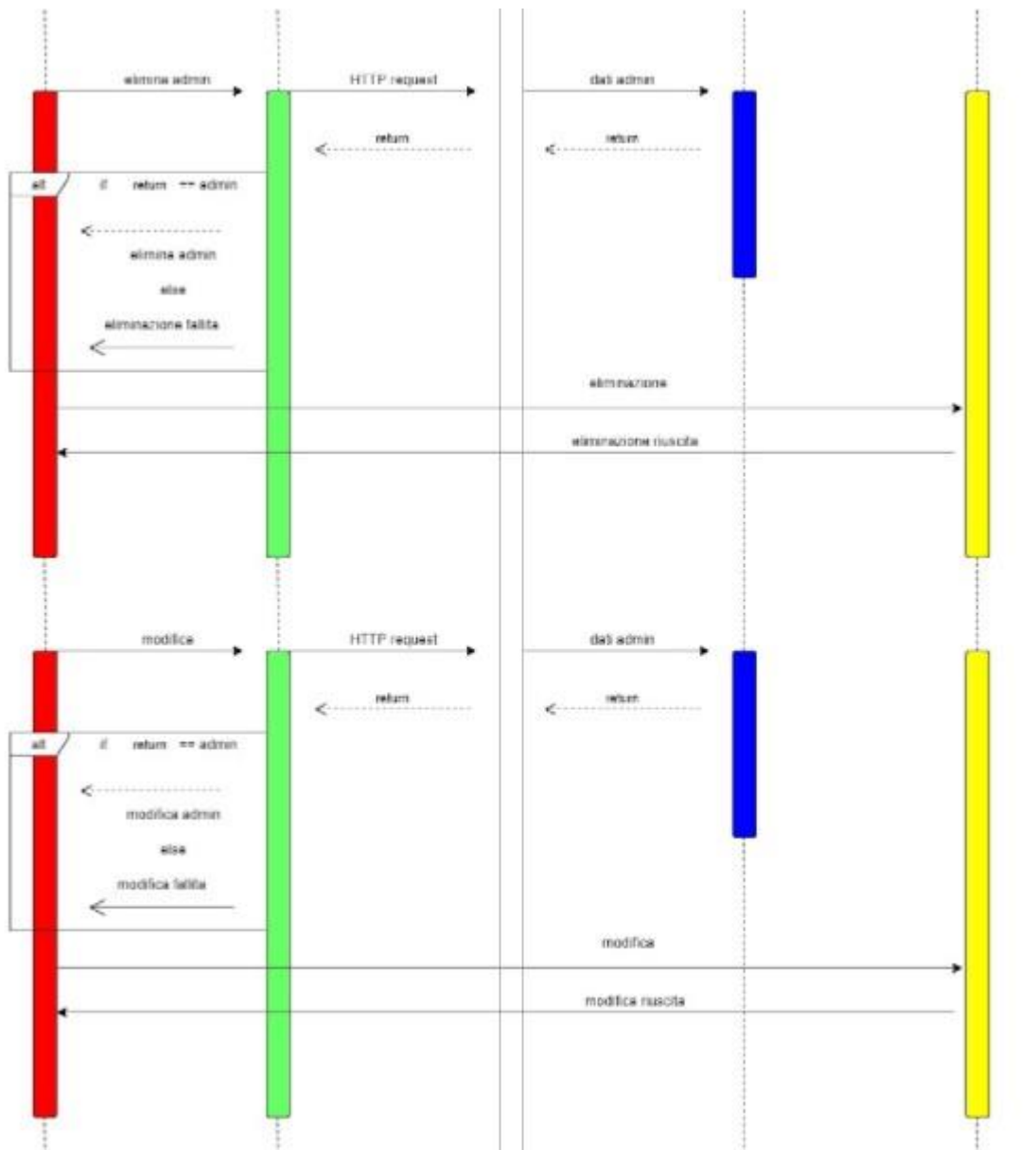
Per prima cosa il nostro admin dovrà effettuare il login, se i dati inseriti saranno corretti l'admin sarà indirizzato nella pagina dedicata. L'admin può accedere al market dove ha varie funzionalità, ovvero, può aggiungere, modificare o cancellare un prodotto, se vi saranno problemi di connessione o le credenziali dell'admin saranno errate in confronto a quello che c'è nel database, verrà mostrato un messaggio di errore.

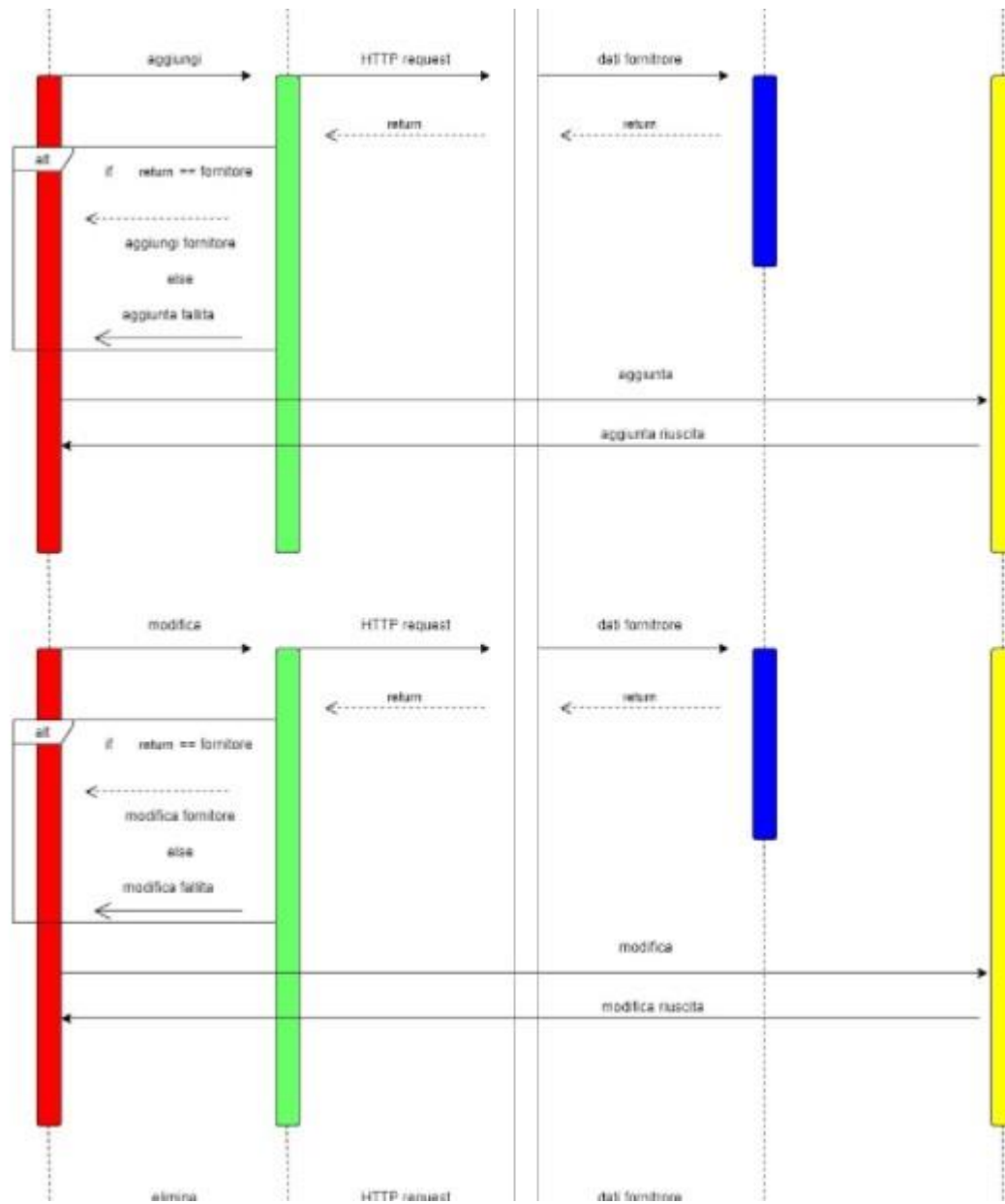
Successivamente l'admin può accedere all'area dedicata al quiz e può decidere se aggiungere o eliminare una domanda. L'admin inoltre può aggiungere, modificare o eliminare altri admin o eventuali fornitori.

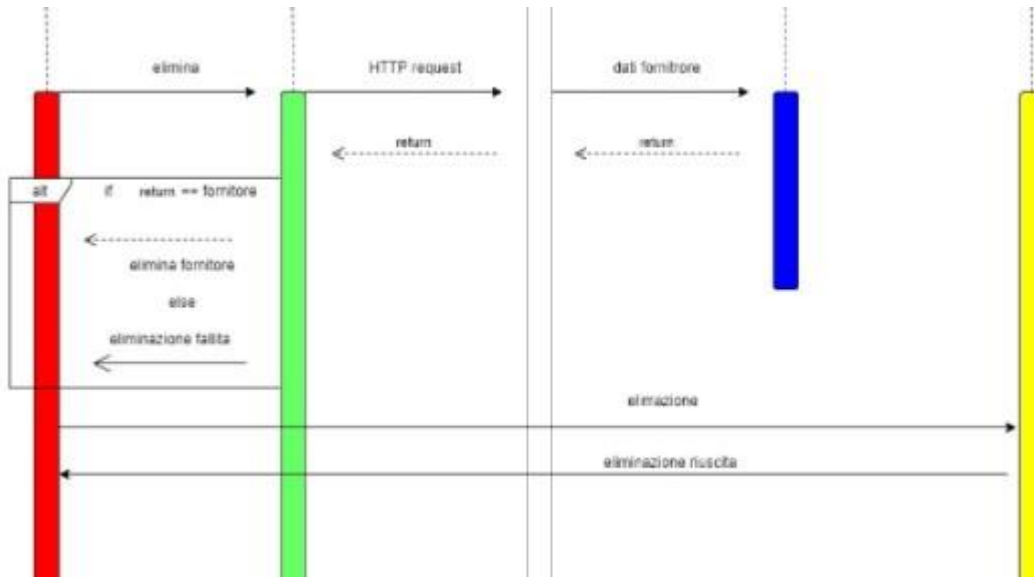
Se nelle funzioni descritte in precedenza, si dovessero riscontrare errori di connessione o campi incompleti, verrà visualizzato un messaggio di errore e l'admin potrà ripetere le operazioni











6.5Security

Affinché i dati dell'utente siano al sicuro, viene utilizzato l'algoritmo **MD5** per la codifica e decodifica della password. Per avere una maggiore sicurezza si concatena alla stringa della password un “**salt**”, in modo da poter contrastare gli attacchi a vocabolario o a forza bruta.

6.6GUI Structure

La struttura della GUI si basa su un insieme di Frame, al cui interno, tramite dei Button ci sposteremo tra di essi. La struttura è abbastanza semplice in quanto ogni Frame viene richiamato solo quando un evento voluto dall'utente lo

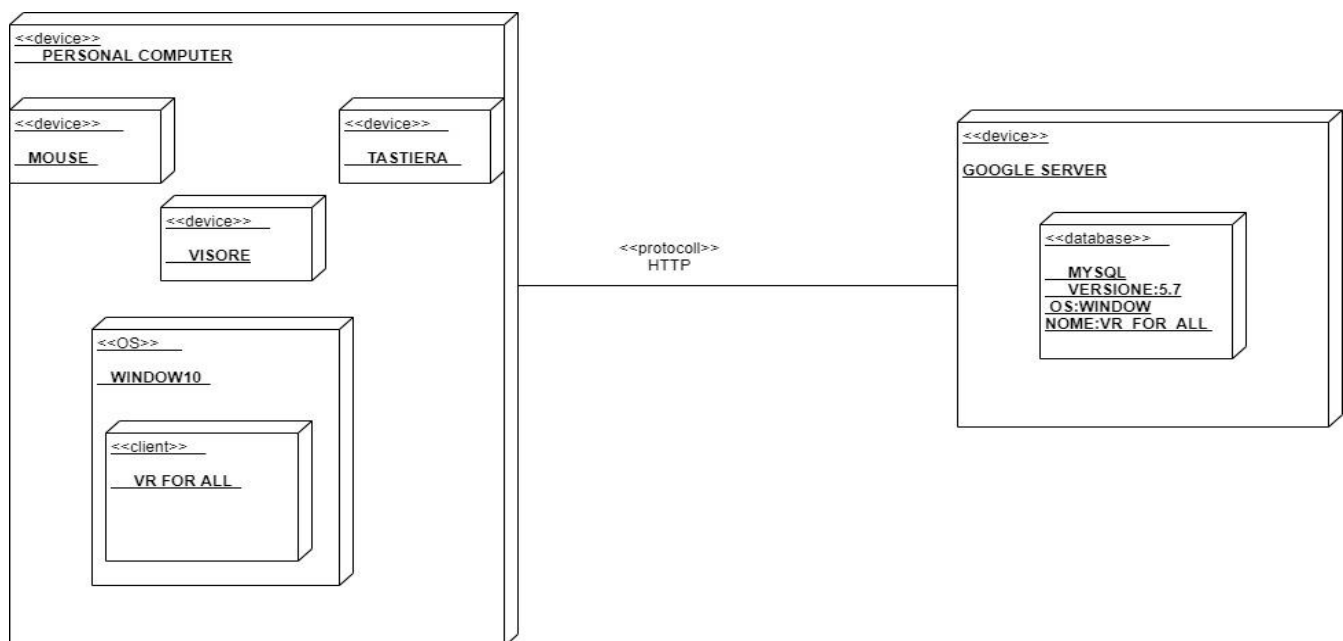
farà comparire. Risulta evidente che ogni Frame viene subordinato dalla DashBoard principale in cui l'utente troverà i Button per operare all'interno di tutto il software(i Frame sono descritti in dettaglio nel paragrafo precedente).

7. Testing Phase

Dopo aver sviluppato le precedenti fasi, ora è possibile passare ad una prima fase di testing di sistema, in particolare in questa fase verrà testato il software utilizzando un approccio **black-box**. Si valuta dunque il software “a scatola chiusa”, senza cioè verificare in questa fase la validità del codice (che è comunque stata testata passo passo durante la fase di sviluppo), ma controllando come il programma finale si comporta in determinate condizioni e verificando che rispetti i requisiti funzionali richiesti. Il software verrà inizialmente testato con un insieme di input “corretti”, per verificare che il risultato soddisfi i requisiti funzionali, infine verranno inseriti degli insiemi di input che tenteranno di verificare il comportamento di quest’ultimo nel caso di input errati.

7.1 DEPLOYMENT DIAGRAM

Il software è composto da una sola applicazione Desktop standalone, quindi il deployment diagram sarà composto da un solo nodo(client), ed un solo nodo per il server.



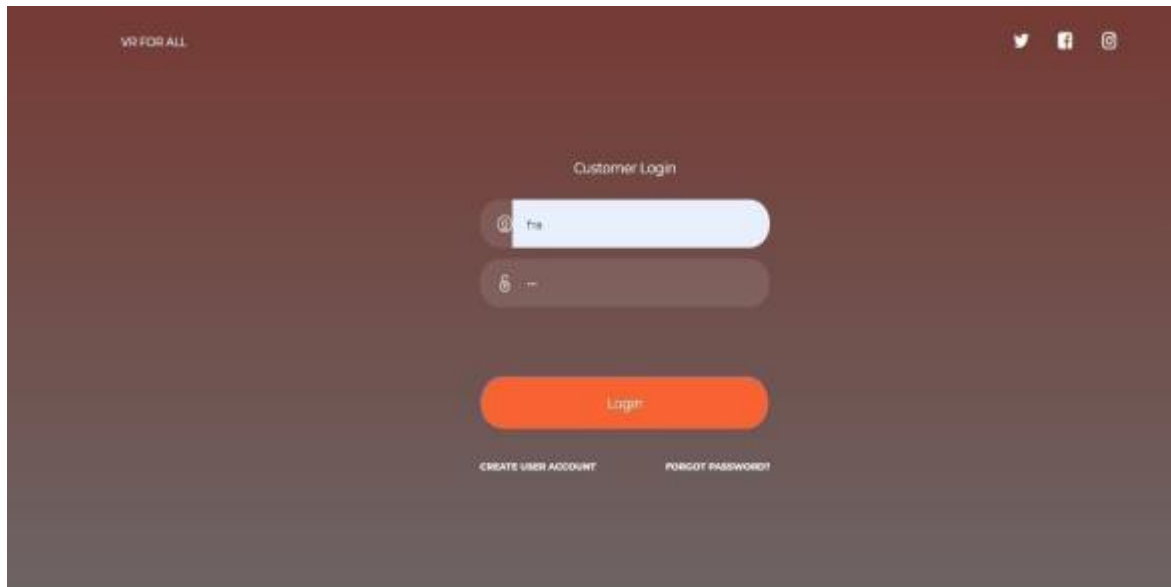
7.2 Validation

In questa fase si verificherà se il software permette di rispondere alle esigenze dell'utente, rispettando i seguenti requisiti funzionali:

RQ1.0: GESTIONE ACCESSO(UTENTE)

RQ1.0.1: Registrazione utente/**RQ1.0.2:** Login/**RQ1.0.3:** Logout

Dopo aver testato l'aggiunta di un nuovo utente, il login e il logout, il tutto risulta perfettamente funzionante ed in linea con i requisiti richiesti. In particolare effettuando l'accesso, si presentano esclusivamente dati ed informazione di quest'ultimo senza che ci siano sovrapposizioni di dati con altri utenti. Nella pagina generale viene fornita all'utente la possibilità di scegliere con quale tipologia di utente effettuare l'accesso. Dunque un utente in alcun modo riesce ad accedere a contenuti altrui. Inoltre è stata testata anche la sicurezza dell'accesso, provando ad accedere con nomi e password errati senza alcuna conseguenza sull'integrità del software. (come illustrato nelle figure sottostanti)



LOGIN PAGE

The image shows a registration form for a service called "VR FOR ALL". The form is set against a dark red background. It includes the following fields and elements:

- Top left: "VR FOR ALL" text.
- Top right: Social media icons for Twitter, Facebook, and Instagram.
- Username field: Labeled "username", containing the text "alejandro".
- Second username field: Labeled "username", containing the text "mad".
- Third username field: Labeled "username", containing the text "ccc".
- Email field: Labeled "email", containing the text "ale@gmail.com".
- Address field: Labeled "address", containing the text "Via Cav.F. Sengioanni 4".
- Phone field: Labeled "phone", containing the text "87145255555".
- First password field: Labeled "password", containing the text "ale".
- Second password field: Labeled "password", containing three dots "...".
- Bottom center: A large orange button labeled "Create account".
- Bottom right: Small copyright text: "© 2021. Designed and Coded by CoderDin".

SUBMIT PAGE

TEST

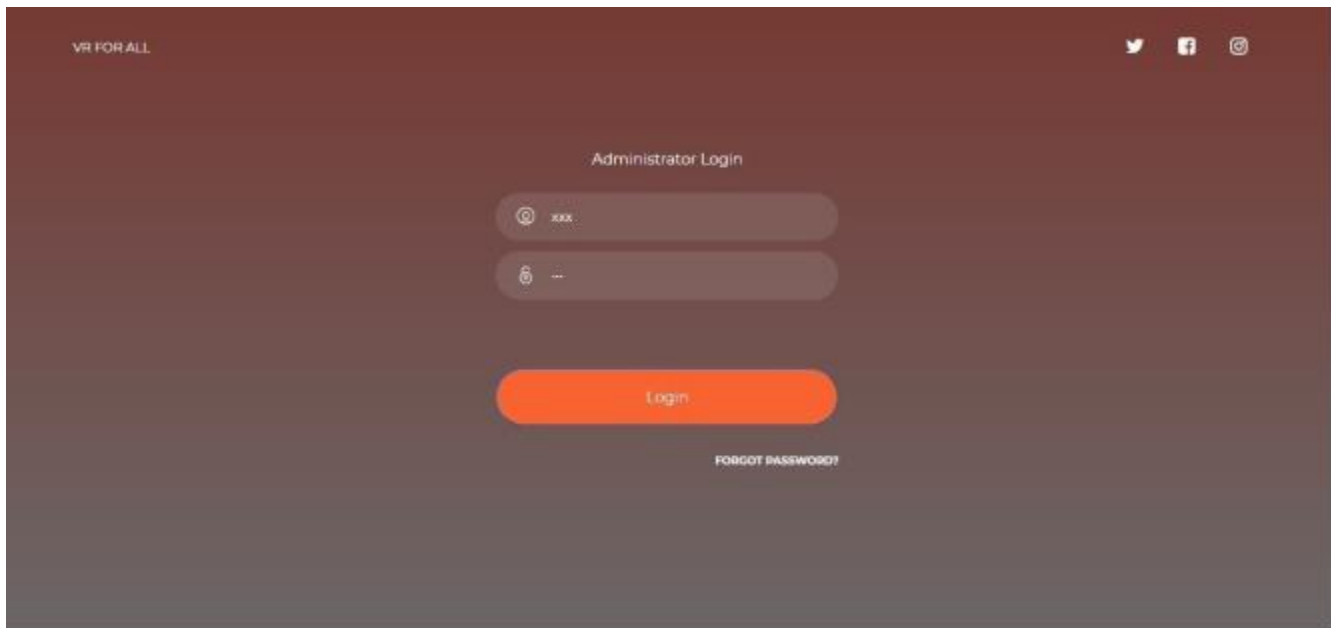
- + Registrazione con input corretti (sequenza principale degli eventi, risultato atteso: registrazione effettuata).
- + Registrazione di un utente già registrato (sequenza alternativa degli eventi, risultato atteso: avviso errore).
- + Login con username e password corretti (sequenza principale degli eventi, risultato atteso: login effettuato).
- + Login con input errati (sequenza alternativa degli eventi, risultato atteso: avviso errore).
- + Login con input non appartenenti a nessun utente registrato (sequenza alternativa degli eventi, risultato atteso: avviso errore).

RQ1.1: GESTIONE ACCESSO (ADMIN)

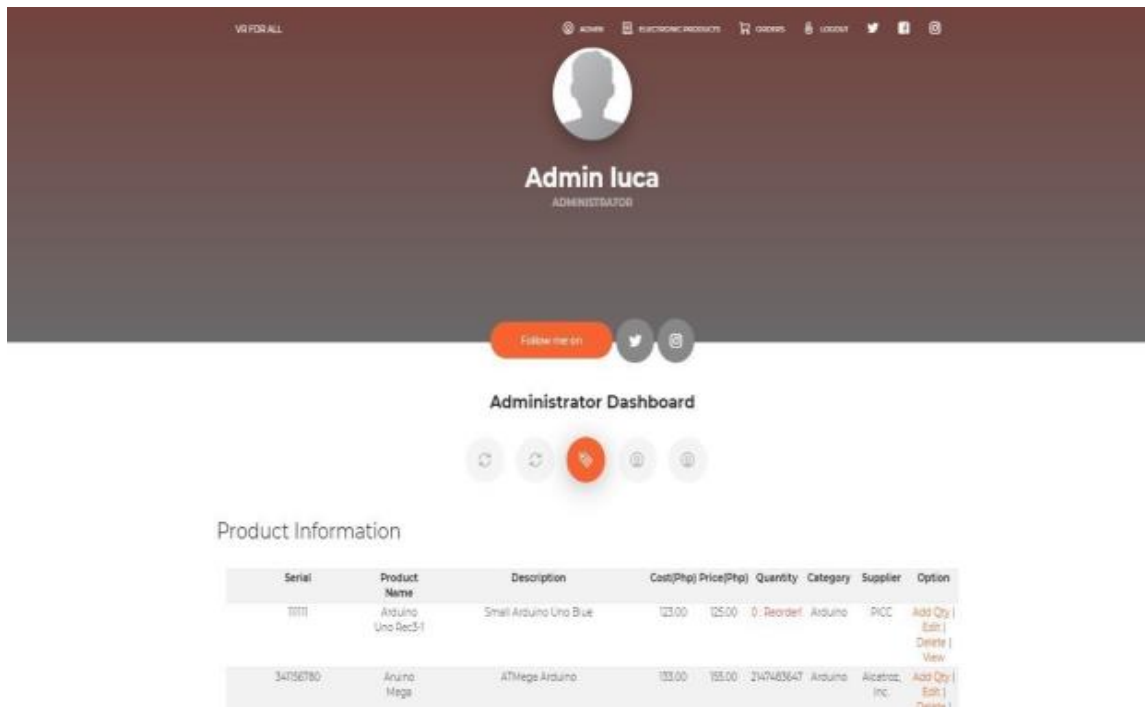
RQ1.1.1: Login/RQ1.1.2: Logout/RQ1.1.2: Profilo

Dopo aver inserito correttamente gli admin nel database , abbiamo testato il login e il logout, il tutto risulta perfettamente funzionante ed in linea con i requisiti richiesti. In particolare effettuando l'accesso, si presentano esclusivamente dati ed informazione di quest'ultimo senza che ci siano sovrapposizioni di dati con altri utenti.

L' admin dopo aver effettuato il login , va nell'area dedicata per il profilo , dove potrà gestire la piattaforma(utenti , prodotti ,fornitori, admin). Nel testare i requisiti in esame non sono stati riscontrati malfunzionamenti.



LOGIN PAGE(ADMIN)



DASHBOARD ADMIN

TEST :

- + Login con input corretti (sequenza principale degli eventi, risultato atteso: Login effettuata).
- + Login con input non appartenenti a nessun admin presente(sequenza alternativa degli eventi, risultato atteso: avviso errore).
- + Aggiornamenti testati su :**prodotti market/admin/fornitori** ,compilando i campi richiesti con input corretti (sequenza principale degli eventi, risultato atteso: update effettuata).
- + Eliminazione :**utente/prodotto/admin fornitori** , compilando i campi richiesti con input corretti (sequenza principale degli eventi, risultato atteso: update effettuata)
- + Aggiornamenti testati su :**prodotti market/admin/fornitori** ,compilando i campi richiesti con input errati(sequenza non principale degli eventi, risultato atteso: errore).
- + Eliminazione :**utente/prodotto/admin fornitori** , compilando i campi richiesti con input errati (sequenza alternativa degli eventi, risultato atteso: errore update)
- + Aggiunta : **prodotto /admin/fornitori/domanda** : ,compilando i campi richiesti con input corretti (sequenza principale degli eventi, risultato atteso: aggiunta nuovo campo effettuata)

Q2.0: UTILIZZO PIATTAFORMA(UTENTE)

RQ2.0.1: Tour Vr(chat compresa)/**RQ2.0.2:** Market/**RQ2.0.3:** Descrizione pianeti/**RQ2.0.4:** Quiz sui pianeti.

Sono state testate tutte le funzionalità del software, soddisfacendo le richieste del cliente. Il market non presenta malfunzionamenti , l'utente dopo aver eseguito l'accesso, può visualizzare i prodotti all'interno, può acquistare un prodotto , visualizzare carrello(come visto precedentemente nella descrizione dei Frame),le funzionalità presenti sono state testate con successo.

Per quanto riguarda l'esperienza in realtà virtuale , sono stati svolti test sia con il visore che senza. Non sono stati riscontrati problemi. Se durante l'utilizzo del software ,il PC non dovrebbe collegarsi in maniera corretta tramite wifi al telefono si consiglia una connessione cablata(i bug di applicazioni esterne non incidono sulla nostra piattaforma).

Abbiamo svolto varie volte il quiz, non si sono presentato malfunzionamenti dal software.

RQ3: GESTIONE PIATTAFORMA(ADMIN)

RQ3.1: Elimina utente/**RQ3.2:** Inserisci admin/**RQ3.3:** Elimina admin/**RQ3.4:**Aggiorna admin/**RQ3.5:** Aggiungi prodotto/**RQ3.5:** Modifica prodotto/**RQ3.5:** Elimina prodotto.

Anche la richiesta del cliente, la quale era che all'interno del software ci fossero 1 o più admin è stata soddisfatta e tastata. Durante la fase di testing non sono state

riscontrate anomalie nella gestione della piattaforma. Tutti i requisiti descritti in precedenza sono stati sviluppati. L'admin può gestire i clienti , prodotti ed admin senza problemi sia di accesso sia delle funzionalità stesse.

Sono stati testati tutti i requisiti funzionali , comprendendo anche i casi in cui il sistema da errore come ad esempio , lasciare campi vuoti.

7.3Verification

È tuttavia necessario ora verificare che il software sia non solo funzionale, ma anche privo di errori. Per questo motivo verranno effettuati dei test con lo scopo di cercare di trovare qualche eventuale errore, inserendo degli insiemi di dati in tutti i possibili input del programma:

TEST #1

test	VALORE DI INGRESSO	RISULTAT O
	Sequenza di caratteri alfabetici	Corretta ricerca
	Sequenza di caratteri speciali	Campo non accettato
	Stringa vuota	Campo non accettato

Questo particolare test è stato ripetuto per tutti i campi di input nei quali era richiesta una stringa ed i risultati sono stati equivalenti, evidenziando dunque come sia necessario fornire un valore in input: nel caso in cui non sia inserito nulla nel campo, la pressione del pulsante di invio non effettuerà alcuna azione, finché l'utente non avrà digitato qualcosa.

CONCLUSIONI

I test condotti non hanno dunque rilevato apparenti casi di malfunzionamenti dovuti ad inserimenti errati di dati di input. La natura intrinseca del programma, tuttavia, fa sì che il funzionamento di quest'ultimo sia fortemente dipendente dal trascorrere del tempo. È dunque possibile l'insorgere di qualche bug nel caso di determinate combinazioni di fattori comprendenti la data odierna, motivo per il quale si renderebbe necessaria una fase di **alpha testing** iniziale.




8.Release &Maintenance

8.1 Release 1.0

Il software prima di essere rilasciato a titolo definitivo, è stato consegnato al cliente finale per effettuare il release testing. Questo test è stato importante per capire che il software ha soddisfatto i requisiti in maniera completa.

Questo ci ha permesso anche di raccogliere informazioni per mantenere il sistema sempre aggiornato. È necessario fare una fase di manutenzione, per cercare di tenere il software più efficace e efficiente possibile anche con il passare del tempo ed evitare che degradi.

Alcune modifiche che possono essere effettuate sul sistema sono :

-  Admin può gestire il tour, avendo la possibilità di gestire i pianeti.
-  L'utente può chattare direttamente con l'admin
-  Spiegazione pianeti nel tour in realtà virtuale

GLOSSARIO:

Architettura: *L'architettura software `e l'organizzazione di base di un sistema, espressa dalle sue componenti, dalle relazioni tra di loro e con l'ambiente, e i principi che ne guidano il progetto e l'evoluzione.*

Cliente: persona o gruppo di persone che richiede il prodotto

IDE: *Ambienti di sviluppo integrati. Forniscono strumenti che supportano il processo di sviluppo del software, inclusi editor per scrivere e modificare programmi e debugger per localizzare errori.*

Ingegneria del software: *Disciplina che studia le modalità e le metodologie con le quali avviene il processo di produzione del software, dalla raccolta dei requisiti fino alla realizzazione del prodotto e al suo rilascio in produzione. Tale disciplina ha pertanto molteplici obiettivi: individuare quali sono nel processo di produzione del software le diverse fasi attraverso le quali tale processo si struttura, il cosiddetto ciclo di vita del software; definire e documentare opportune metodologie di sviluppo che possano coprire le fasi individuate e identificare le tecnologie più appropriate per realizzare i prodotti utilizzando le suddette metodologie. Tali obiettivi vengono perseguiti considerando il processo di sviluppo del software da differenti angolazioni e utilizzando approcci complementari: non solo un approccio scientifico e tecnologico, ma anche un approccio di tipo economico e organizzativo, che sia in grado di evidenziare costi e benefici nel processo produttivo e che giustifica appieno il termine ingegneria del software assegnato a tale disciplina.*

Panoramica software: con questa parola s'intende una visione non troppo tecnicistica di quello che sarà il prodotto software. Vengono introdotte le caratteristiche del software senza entrare troppo nel dettaglio.

Prodotto: *è il software che vogliamo realizzare Normalmente quando si cerca di pensare al software (inteso come il prodotto di un certo processo), viene da paragonarlo alla produzione di un qualche oggetto.*

Panoramica software: *con questa parola s'intende una visione non troppo tecnicistica di quello che sarà il prodotto software. Vengono introdotte le caratteristiche del software senza entrare troppo nel dettaglio.*

Software: *software termine inglese che indica l'insieme dei programmi necessari a un elaboratore elettronico per il suo funzionamento e per la soluzione di problemi.*

Il software spesso è dipendente dalle caratteristiche dell'hardware del calcolatore stesso, ossia dall'insieme dei suoi componenti fisici (apparecchiature, circuiti, dispositivi ecc.). Sinonimi: prodotto, sistema.

Standard: *Modello, tipo, norma, riferimento cui si devono uniformare, o a cui sono conformi, i prodotti e i procedimenti.*

UML: *Unified Modeling Language. Si rimanda al paragrafo 2 per ulteriori chiarimenti.*

XAMPP: *XAMPP è una piattaforma software multiplatforma e libera costituita da Apache HTTP Server, il database MariaDB e tutti gli strumenti necessari per utilizzare i linguaggi di programmazione PHP e Perl. Il nome è un acronimo dei software sopra citati (la X sta per x- platform, l'abbreviazione di cross-platform in lingua inglese ovvero multiplatforma).*

Stakeholder: *Tutti i soggetti, individui od organizzazioni, attivamente coinvolti in un'iniziativa economica (progetto, azienda), il cui interesse è negativamente o positivamente influenzato dal risultato dell'esecuzione, o dall'andamento, dell'iniziativa .*

Autenticazione: *L'autenticazione è il processo attraverso il quale viene verificata l'identità di un utente che vuole accedere ad un computer o ad una rete. È il sistema che verifica, effettivamente, che un individuo è chi sostiene di essere. L'autenticazione è diversa dall'identificazione (la determinazione che un individuo sia conosciuto o meno dal sistema) e dall'autorizzazione (il conferimento ad un utente del diritto ad accedere a specifiche risorse del sistema, sulla base della sua identità).*

Requisiti funzionali: *descrivono i servizi, o funzioni, offerti dal sistema normalmente attivati da user-inputs.*

Requisiti non funzionali: *descrivono vincoli sui servizi offerti dal sistema, e sullo stesso processo di sviluppo.*

Skill: *conoscenze e abilità che qualcuno possiede. Possono essere acquisite attraverso il processo di istruzione, l'addestramento, l'esperienza lavorativa o essere semplicemente capacità innate.*

Rischi di progetto: *I rischi di progetto sono legati al compimento del progetto ad esempio un rischio potrebbe far slittare la consegna del software.*

Rischi di prodotto: *Rischi che generano un problema funzionale al prodotto software.*

Tabella dei rischi: *Tale tabella si basa, oltre che sul catalogo dei rischi, anche su una strategia suggerita dalla U.S. Air Force Questa strategia vuole che il capo progetto individui i fattori di rischio che incidono*

sui componenti di rischio del software: prestazioni, costi, supporto e tempi.

Requisiti informali: *Le specifiche informali fanno uso di linguaggio naturale per descrivere i requisiti. La sintassi e la semantica non sono formalmente definite.*

Stakeholder: *Tutti i soggetti, individui od organizzazioni, attivamente coinvolti in un'iniziativa economica (progetto, azienda), il cui interesse è negativamente o positivamente influenzato dal risultato dell'esecuzione, o dall'andamento, dell'iniziativa e la cui azione o reazione a sua volta influenza le fasi o il completamento di un progetto o il destino di un'organizzazione. Nell'ambito di un progetto, sono i soggetti relativi al cliente, al fornitore, i membri del team di progetto, i*

fruitori dei risultati in uscita dal progetto, i gruppi di interesse locali relativamente all'ambiente dove il progetto si sviluppa e l'azienda opera. Tra gli stakeholder, vi sono i soggetti senza i quali l'impresa non sopravvive. L'identificazione degli s. si ottiene mediante un elenco casuale e libero dei soggetti coinvolti nel progetto (tecniche di brainstorming) oppure mediante liste di controllo descrittive dell'ambiente di progetto o di progetti precedenti (check list) o infine mediante simulazioni dell'ambiente di progetto per rintracciare gli stakeholder interni ed esterni (rappresentazione). Sinonimi: cliente, utente.

Casi d'uso: *è una tecnica usata nei processi di ingegneria del software per effettuare in maniera esaustiva e non ambigua, la raccolta dei requisiti al fine di produrre software di qualità.*

GUI: *acronimo. Graphical User Interface in informatica è un tipo di interfaccia utente che consente l'interazione uomo-macchina in modo visuale utilizzando rappresentazioni grafiche.*

Diagramma dei casi d'uso: *Il diagramma dei casi d'uso assegna i casi d'uso agli attori ed evidenzia le relazioni tra i casi d'uso. Nel diagramma dei casi d'uso un attore può essere rappresentato da un omino stilizzato, con sotto il nome dell'attore (normalmente rappresenta un attore esterno), Un caso d'uso è rappresentato come un ellissi con all'interno il nome del caso d'uso. Gli attori sono associati ai casi d'uso per mezzo di frecce che puntano dall'attore al caso d'uso. Oltre a questo è possibile la generalizzazione degli attori e dei casi d'uso.*

Diagramma delle attività: *Si prendono le descrizioni dei flussi: sia quello principale che quelli alternativi sono fusi nello stesso diagramma per poter far vedere l'insieme delle azioni compiute dal caso d'uso preso in considerazione.*

Per preparare un diagramma delle attività si disegna un cerchio pieno che indica il punto di inizio dell'attività; quando un caso d'uso viene attivato, si ci sposta sulla prima attività (rappresentata da un rettangolo dai bordi arrotondati), quando l'attività ha compiuto il suo compito, si ci sposta alla seconda e così via fino allo stato finale (rappresentato da un cerchio pieno inscritto in un secondo cerchio vuoto) in cui il caso d'uso termina. Nel momento in cui bisogna fare una scelta (e quindi passare ad un flusso alternativo) il collegamento avviene attraverso un rombo che permette di spezzare la linea di

congiunzione tra le attività a seconda delle condizioni specificate su ogni ramo (sono racchiuse tra parentesi quadre); quando i flussi alternativi si riuniscono lo fanno attraverso un secondo rombo.

Diagramma di sequenza: *I diagrammi di sequenza, comunemente usati dagli sviluppatori, modellano le interazioni tra gli oggetti in un unico caso d'uso. Essi illustrano come le diverse parti di un sistema interagiscono tra loro per svolgere una funzione, e l'ordine in cui le interazioni avvengono quando un particolare caso d'uso viene eseguito. In parole più semplici, un diagramma di sequenza mostra*

diverse parti del lavoro di un sistema in una 'sequenza' per ottenere qualcosa.

Query: *In informatica, interrogazione di un database per estrarre o aggiornare i dati che soddisfano un certo criterio di ricerca.*