**­KEY-KNOWLEDGE**



**Nome Documento**

Problem Statement Version 1.4.2

SOMMARIO

[Scenari: 6](#_Toc57812783)

[1.L’utente utilizza il sistema per la prima volta: 6](#_Toc57812784)

[Registrazione corretta[MK1] 8](#_Toc57812785)

[Registrazione non corretta[MK2] 8](#_Toc57812787)

[Pop-up avviso email[MK3] 9](#_Toc57812788)

[2.L’utente fa il login: 9](#_Toc57812790)

[Login corretto[MK4] Login non corretto[MK5] 9](#_Toc57812791)

[3.Tutorial: 10](#_Toc57812792)

[Regole gioco[MK6] 10](#_Toc57812793)

[Risposta domanda[MK7] 11](#_Toc57812794)

[Scelta domanda[MK8] 12](#_Toc57812795)

[Creazione domanda[MK9] 12](#_Toc57812796)

[Schermata principale[MK10] 13](#_Toc57812797)

[Schermata KNOWLEDGE[MK11] 14](#_Toc57812798)

[4.L’utente gioca in modalità classica: 14](#_Toc57812799)

[Scelta modalità[MK12] 14](#_Toc57812800)

[Caricamento[MK13] 15](#_Toc57812801)

[Caricamento completato[MK14] 15](#_Toc57812802)

[Caricamento fallito[MK15] 16](#_Toc57812803)

[Esempio domanda[MK16] 16](#_Toc57812804)

[Esempio domanda[MK17] 17](#_Toc57812805)

[Attesa fine partita[MK18] 17](#_Toc57812806)

[Fine partita[MK19] 18](#_Toc57812807)

[5.L’utente gioca in modalità MISC: 18](#_Toc57812808)

[Attesa domanda[MK20] 19](#_Toc57812809)

[Risposta domanda con Report[MK21] 19](#_Toc57812810)

[6.L’UserManager accede all’area di Management: 19](#_Toc57812811)

[User-Management Select[MK22] 20](#_Toc57812812)

[Management Area[MK23] 20](#_Toc57812814)

[Delete User[MK24] 21](#_Toc57812815)

[Stats User[MK25] 21](#_Toc57812816)

[View Report[MK26] 22](#_Toc57812817)

[7.Il Question Manager accede all’area di Management 22](#_Toc57812818)

[Enter Question Manager[MK27] 23](#_Toc57812819)

[Question Area[MK28] 23](#_Toc57812820)

[Adding Category[MK29] 24](#_Toc57812821)

[New Category[MK30] 24](#_Toc57812822)

[View Questions[MK31] 25](#_Toc57812823)

[Adding Question[MK32] 26](#_Toc57812824)

[New Question[MK33] 26](#_Toc57812825)

[Deleting Category[MK34] 27](#_Toc57812826)

[Update password [MK35] 27](#_Toc57812827)

[3.Requisiti funzionali 28](#_Toc57812828)

[[FR01]: 28](#_Toc57812829)

[[FR02]: 28](#_Toc57812830)

[[FR03]: 28](#_Toc57812831)

[[FR04]: 28](#_Toc57812832)

[[FR05]: 28](#_Toc57812833)

[[FR06]: 29](#_Toc57812834)

[[FR07]: 29](#_Toc57812835)

[[FR08]: 29](#_Toc57812836)

[[FR09]: 29](#_Toc57812837)

[[FR10]: 29](#_Toc57812838)

[[FR11]: 29](#_Toc57812839)

[[FR12]: 29](#_Toc57812840)

[Tabella delle priorità: 30](#_Toc57812841)

[4.Requisiti non funzionali 30](#_Toc57812842)

[[NFR01]: 31](#_Toc57812843)

[[NFR02]: 31](#_Toc57812844)

[[NFR03]: 32](#_Toc57812845)

[[NFR04]: 32](#_Toc57812846)

[[NFR05]: 32](#_Toc57812847)

[[NFR06]: 32](#_Toc57812848)

[[NFR07]: 32](#_Toc57812849)

[[NFR08]: 33](#_Toc57812850)

[[NFR9]: 33](#_Toc57812851)

[5.Ambiente di destinazione: 34](#_Toc57812852)

[6.Scadenze: 34](#_Toc57812853)

[7.Criteri di accettazione dei test: 34](#_Toc57812854)

# ****TABELLA REVISIONI****

|  |  |  |  |
| --- | --- | --- | --- |
| **DATA** | **VERSIONE** | **DESCRIZIONE** | **AUTORE** |
| **17/10/2020** | **1.0** | **Prima versione del problemStatemant** | **Crescenzo Manzone**  **Franco Nicola Fernando**  **Giovanni Battista Mercurio** |
| **23/10/2020** | **1.1** | **Modifica scenari** | **Crescenzo Manzone** |
| **24/10/2020** | **1.2** | **Modifica Requisiti Funzionali** | **Franco Nicola Fernando** |
| **25/10/2020** | **1.3** | **Modifica Requisiti non Funzionali** | **Giovanni Battista Mercurio** |
| **27/10/2020** | **1.4** | **Unione delle modifiche effettuate** | **Crescenzo Manzone**  **Franco Nicola Fernando**  **Giovanni Battista Mercurio** |
| **6/11/2020** | **1.4.2** | **Revisione requisiti** | **Crescenzo Mazzone** |

# Problem Statement

# ****Dominio del problema****

**Il software dovrà fornire la possibilità di interagire con altri utenti e sfidarsi in un quiz. Il vincitore sarà il giocatore che risponde al maggior numero di domande, generate da un agente intelligente che ha la particolarità di evolversi e adattarsi all’utente. Le domande sono suddivise in categorie (STORIA, GEOGRAFIA, SCIENZE, ARTE, CULTURA GENERALE). Una singola partita è sempre 1v1.**

# ****Scenari:****

## **1.L’utente utilizza il sistema per la prima volta:**

**Giovanni Alfonsi scarica l’applicazione e la apre per la prima volta. Gli appare davanti la schermata di registrazione che riempie con i suoi dati. Dopo aver inserito correttamente i suoi dati Giovanni clicca sul pulsante SIGN-IN.**

**Nel mok-up MK1 vediamo Giovanni che ha riempito il form con dati corretti.**

**Gli appare un pop-up.**

**Giovanni va nella sua casella per confermare l’account, torna sull’applicazione per effettuare il login. Nel mok-up MK3 vediamo un esempio di inserimento dati non corretto e nel mok-up MK2 come il sistema avvisa l’utente dei dati non corretti.**

## **Registrazione corretta[MK1]**

## 

## **Registrazione non corretta[MK2]**



## Pop-up avviso email[MK3]

## 

## **2.L’utente fa il login:**

**Giovanni inserisce correttamente i suoi dati e clicca sul pulsante LOG-IN. Essendo il primo utilizzo dell’app da parte di Giovanni, dopo il click su LOG-IN, partirà il tutorial.**

**Nel mok-up MK4 un esempio di inserimento dati corretto.**

**Nel mok-up MK5 un esempio di inserimento dati non corretto**

## **Login corretto[MK4] Login non corretto[MK5]**

****

## **3.Tutorial:**

**Giovanni si trova davanti una schermata dove gli vengono esposte le regole del gioco.**

**L’unica azione che Giovanni può fare per andare avanti è cliccare su NEXT, quindi clicca su NEXT, che lo fa proseguire con il tutorial. Nel mok-up MK6 abbiamo un esempio.**

## **Regole gioco[MK6]**



**Giovanni si trova davanti una domanda a cui deve rispondere con il tempo che scorre, gli viene indicato di cliccare su una risposta e successivamente su CONFIRM. Quindi Giovanni clicca su una risposta e poi su CONFIRM, che lo fa proseguire con il tutorial.**

**Nel mok-up MK7 abbiamo un esempio. In particolare in questo esempio dopo che Giovanni clicca su CONFIRM la domanda che ha selezionato di colorerà di verde perché ha risposto correttamente. Se risponde in maniera non corretta si colora di rosso.**

## **Risposta domanda[MK7]**



**Giovanni si trova davanti una domanda da scegliere, prima di fare qualsiasi azione, gli appare un pop-up dove gli spiega che questa è una domanda che lui sceglie e che viene usata per il suo pseudo-avversario, visto che questo è un tutorial non ha un avversario. Clicca su NEXT per scegliere la domanda.**

**Giovanni sceglie la tipologia, dopodiché sceglie una delle 4 domande relative alla tipologia scelta e infine clicca su CONFIRM. Nel mok-up MK8 abbiamo un esempio. Giovanni va avanti con il tutorial.**

## **Scelta domanda[MK8]**



**Giovanni si trova davanti una schermata dove deve scrivere una domanda. Prima di fare qualsiasi azione gli appare un pop-up che gli spiega che la domanda che lui scriverà sarà fatta rispondere dal suo pseudo-avversario. Giovanni scrive la domanda e le risposte e sceglie anche la domanda giusta, dopodiché può confermare andando cosi avanti con il tutorial. Nel mok-up MK9 abbiamo un esempio.**

## **Creazione domanda[MK9]**



**Giovanni si trova nella schermata principale dove gli vengono spiegate le azioni che può fare. Il tutorial gli farà apparire i messaggi in 7 passaggi arrivando alla schermata di KNOWLEDGE, qui sarà spiegato l’utilità di questa pagina e il funzionamento dell’IA.**

**Nel mok-up MK10 un esempio di schermata principale, e nel mok-up MK11 un esempio di schermata di KNOWLEDGE.**

## **Schermata principale[MK10]**



## **Schermata KNOWLEDGE[MK11]**



**Giovanni legge il funzionamento e clicca su HO CAPITO, il tutorial termina e ritorna alla schermata principale.**

## **4.L’utente gioca in modalità classica:**

**Giovanni ha terminato il tutorial e crede di essere in grado di affrontare una partita reale con un altro giocatore. Trovandosi nella schermata principale sceglie la modalità di gioco classica e clicca su play. Nel mok-up MK12 un esempio di scelta della modalità di gioco. A Giovanni appare una schermata di caricamento della partita, nel mok-up MK13 abbiamo un esempio. Alla fine del caricamento viene abbinato con un utente con nickname SOLVERMASTER, nel mok-up MK14 abbiamo un esempio.**

## **Scelta modalità[MK12]**



## **Caricamento[MK13]**



## **Caricamento completato[MK14]**



**I due utenti cominciano la partita. Nel mok-up MK15 un esempio di schermata di caricamento(abbinamento) fallito.**

**Giovanni si trova davanti una domanda da rispondere e contemporaneamente anche il suo avversario risponde a una domanda. Nel mok-up MK16 la domanda di Giovanni e nel mok-up MK17 quella del suo avversario.**

**Giovanni risponde velocemente a tutte e 10 le domande, ma il suo avversario sta ancora rispondendo. Nel mok-up MK18 Giovanni che ha finito ma attende, nel mok-up MK19 la schermata di terminazione della partita. Dopo di che Giovanni viene mandato sulla schermata principale.**

## **Caricamento fallito[MK15]**



## **Esempio domanda[MK16]**



## **Esempio domanda[MK17]**



## **Attesa fine partita[MK18]**



## **Fine partita[MK19]**



## **5.L’utente gioca in modalità MISC:**

**Giovanni si trova sulla schermata principale e vuole fare un’altra partita, ma stavolta nella modalità MISC. Quindi clicca prima su MISC MODE e successivamente su PLAY. Raggiunge la schermata di caricamento e viene abbinato con un utente che ha nickname Alex1996 e iniziano a giocare. Alla prima domanda risponde Giovanni per primo. Nel mok-up MK20 ciò che appare a Giovanni dopo che ha risposto alla domanda.**

**Questo meccanismo di attesa permane per ogni domanda in questa modalità.**

**I due giocatori rispondono entrambi alle prime 6 domande, alle domande 7 e 8 scelgono una domanda per il loro avversario e successivamente rispondono alle domande che l’avversario ha scelto per loro, alle domande 9 e 10 creano una domanda e successivamente rispondono alle domande che l’avversario ha creato per loro. Nel rispondere alla domanda 9 Giovanni preme per sbaglio il pulsante di REPORT, gli appare un pop-up per confermare, clicca su NO e risponde alla domanda.**

**Nel mok-up MK21 abbiamo un esempio.**

**Alla fine della partita Giovanni torna alla schermata principale e chiude il gioco.**

## **Attesa domanda[MK20]**



## **Risposta domanda con Report[MK21]**



## **6.L’UserManager accede all’area di Management:**

**L’UserManager Enrico accede all’applicazione facendo login. Inserisce come nickname Rick00 e come password Trump20/. Nella schermata principale ha un tasto UserManagement per entrare nell’area di Management. Enrico clicca su UserManagement ed entra nell’area di Management. Nel mok-up MK22 abbiamo un esempio della schermata principale e nel mok-up MK23 un esempio dell’area di management. Clicca su Users e visualizza la lista degli utenti del sistema con nickname e ultimo accesso. Elimina l’utente Dred65 cliccando su DELETE si apre un pop-up di avviso e conferma. Nel mok-up MK24 abbiamo un esempio. Poi Enrico clicca su STATS di User0 .Nel mok-up MK25 abbiamo un esempio. Clicca su Reports, visualizza l’unico report presente e sospende l’utente che ha creato la domanda segnalata, anche qui gli si apre un pop-up di avviso e conferma. Nel mok-up MK26 abbiamo un esempio. Enrico esce dall’applicazione.**

## User-Management Select[MK22]

## 

## Management Area[MK23]



## Delete User[MK24]



## Stats User[MK25]



## View Report[MK26]



## **7.Il Question Manager accede all’area di Management**

Il Question Manager Alfonso accede all’applicazione facendo login. Inserisce come nicknam Alf33 e come password Dolfin56(. Nella schermata principale ha un tasto QuestionManagement per entrare nell’area di management. Nel mok-up MK27 la schermata principale. Alfonso clicca sul tasto ed entra nell’area di management. Nel mok-up MK28 l’area di management. Visualizza la lista delle categorie e clicca su ADD CATEGORY, nel mok-up MK29 la schermata che gli appare dopo che ha cliccato. Aggiunge una nuova categoria ANIMALS, nel mok-up MK30 la nuova categoria aggiunta. Clicca su VIEW visualizzando cosi la lista di domande della categoria ANIMALS, nel mok-up MK31 la schermata di esempio. Clicca su ADD QUESTION, nel mok-up MK32 la schermata di creazione della domanda. Alfonso clicca su CONFIRM e torna nella lista delle domande di ANIMALS e vede la nuova domanda creata in lista, un esempio nel mok-up MK33. Torna indietro cliccando su BACK TO CATEGORIES. Per sbaglio clicca sul tasto DELETE della categoria HISTORY, gli appare un pop-up di conferma e clicca su NO, nel mok-up MK34 il pop-up. Alfonso chiude l’applicazione.

## Enter Question Manager[MK27]



## Question Area[MK28]



## Adding Category[MK29]



## New Category[MK30]



## View Questions[MK31]



## Adding Question[MK32]

## New Question[MK33]



## Deleting Category[MK34]



## Update password [MK35]



# 3.Requisiti funzionali

## [FR01]:

GESTIONE UTENTE (sign-in)

-Il sistema deve fornire all’utente la possibilità di registrarsi con queste credenziali:

. E-MAIL

. PASSWORD

. NOME

. COGNOME

. NICKNAME

. FOTO (opzionale)

Oppure:

. Sign with facebook-account

. Sign with Google-account

## [FR02]:

GESTIONE UTENTE (confirm)

-Il sistema deve fornire la possibilità di confermare l’account, tramite conferma via e-mail.

## [FR03]:

GESTIONE UTENTE (log-in)

-Il sistema deve fornire all’utente la possibilità di accedere alle funzionalità dell’app e quindi poter sfidare altri utenti registrati inserendo:

-E-MAIL/NICKNAME

-PASSWORD

## [FR04]:

GESTIONE UTENTE (profile management)

-Il sistema deve fornire all’utente la possibilità di visionarie le proprie statistiche (domande sbagliate/vittorie/sconfitte…)

## [FR05]:

GESTIONE UTENTE (update user)

-Il sistema deve fornire all’utente la possibilità di modificare i propri dati personali

## [FR06]:

GESTORE UTENTI (show users)

-Il sistema fornisce all’utente admin, la possibilità di visionare l’username, la foto profilo e le statistiche di ogni utente registrato (Partite vinte, partite giocate e KNOWLEDGE LEVEL).

## [FR07]:

GESTORE UTENTI (remove users)

-Il sistema fornisce all’utente admin, la possibilità di rimuovere utenti inattivi per molto tempo o sospendere utenti segnalati. Le sospensioni durano 1 settimana.

## [FR08]:

GESTORE CATEGORIE (update categories)

-Il sistema deve fornire all’utente admin, la possibilità di modificare o aggiungere nuove modalità di gioco specificando le caratteristiche necessarie della nuova modalità.

## [FR09]:

UTENTE-GIOCATORE (start match)

-Il sistema deve fornire all’utente la possibilità di avviare una sfida contro un altro giocatore, scegliendo opportunamente la modalità di gioco preferita prima di cominciare.

## [FR10]:

UTENTE-GIOCATORE (end match)

-Il sistema deve fornire all’utente la possibilità di interrompere un match con la conseguente rimozione dei dati di quella partita.

[FR11]:

UTENTE-GIOCATORE (playing)

-Il sistema deve fornire all’utente, a seconda della modalità di gioco scelta, la possibilità di: scegliere la risposta corretta, scegliere la domanda da porre all’avversario e di scrivere una domanda per l’avversario.

## [FR12]:

UTENTE-GIOCATORE (report)

-Il sistema deve fornire all’utente la possibilità di segnalare altri utenti , nel caso in cui questi per esempio scrivano testi scurrili all’interno delle domande che scrivono in partita.

# Tabella delle priorità:

(U): utente

|  |  |
| --- | --- |
| **Requisito** | **Livello di priorità** |
| (U)sign-in | Alto |
| (U)Log-in | Alto |
| (U)Profile management | Medio |
| (U)Update-user | Medio |
| (GU)Show users | Medio |
| (GU)Remove users | Medio |
| (GC)Update categories | Basso |
| (UG)Start-match | Alto |
| (UG)End-match | Medio |
| (UG)Playing | Alto |
| (U)Confirm | Medio |
| (UG)Report | basso |
| (U)external log-in | Medio |

(GC): gestore categorie

(GU): gestore utenti

(UG): utente giocatore

# 4.Requisiti non funzionali

1. **Usability**

## [NFR01]:

Level of expertise

Esempio di interfaccia utente:

Se l’utente ruota il dispositivo l’interfaccia non cambierà ne ruoterà.



## [NFR02]:

outcome answers:

**L’utente deve avere la possibilità di vedere se ha risposto in maniera corretta o meno ad una specifica domanda. Lo scopo del nostro sistema è di spronare l’utente a informarsi su quelli che sono gli argomenti più difficoltosi per lui. Alla fine del quiz l’utente saprà a quante domande ha risposto correttamente e potrà vedere tutte le domande a cui ha risposto nel quiz.**

1. **Reliability**

## [NFR03]:

Availability

Il gioco una volta installato è operativo in qualsiasi momento, 24h/24, tranne in ore specifiche di stallo utilizzate per poter controllare, aggiornare il sistema o risolvere errori di vario tipo, ma notificate all’utente tramite schermata di “work in progress”.

## [NFR04]:

Robustness (errors and exceptions) Se si presentano anomalie che interrompono il match, l’utente visionerà una schermata di errore e sarà trasportato alla schermata di inizio e la partita non verrà contata.

1. **Performance**

## [NFR05]:

Il sistema ha una rapidità di esecuzione: il giocatore, dopo aver risposto alla domanda, conosce subito il risultato corrente in base alla sua scelta.

Il sistema supporterà un alto numero di utenti simultanei.

1. **Supportability**

## [NFR06]:

Maintainability

Al lancio del Software nei primi mesi di funzionamento prevediamo delle revisioni del sistema, dato che durante svariati utilizzi possono uscire bug non visionati. in caso di modifiche grafiche o di codice tramite Android studio possono essere apportate velocemente. Per poi mettere online una nuova versione dell’applicazione.

1. **Implementation**

## [NFR07]:

L’applicazione sarà implementata usando linguaggio **JAVA**

IDE utilizzata: **Android Studio**

1. **Constraints**

## [NFR08]:

Additional Documentation

**Regole del quiz:**

**In ogni partita l’utente dovrà rispondere a una serie di domande, in alcuni casi potrà sceglierle per l’avversario o anche crearle per l’avversario. In ogni caso ogni domanda da rispondere, scegliere o creare ha un tempo di azione (intorno ai 30 secondi). Se il tempo di azione termina prima che l’utente ha fatto una scelta gli sarà assegnata una penalità. Non influirà sullo storico poiché non ha dato nessuna risposta. Le penalità si accumulano, ognuna conta -0,25. Le risposte corrette segnano 1 punto e le risposte sbagliate 0. Vince l’utente che fa più punti, in caso di pareggio chi ha il tempo minore. Se l’utente guadagna 2 penalità, alla terza viene messo offline e il suo avversario vince. Oltretutto in base alla modalità di gioco scelta cambieranno i tempi di attesa.**

**Nella modalità classica e nella modalità RESTART gli utenti risponderanno in maniera autonoma alle domande e quando avranno finito di rispondere dovranno attendere l’avversario o viceversa per conoscere il risultato della partita. Nella modalità MISC ad ogni domanda da rispondere, scegliere o creare l’utente dovrà aspettare il suo avversario o viceversa.**

**Funzionamento Intelligenza Artificiale:**

**L’obiettivo dell’intelligenza sarà quello di mettere in difficoltà l’utente proponendogli domande sempre più difficili. Ogni volta che l’utente risponde a una domanda, il suo risultato viene usato dall’intelligenza per capire su quali argomenti è più o meno bravo l’utente. Se un utente risponde in maniera corretta a diverse domande di storia sarà meno probabile che l’intelligenza gli proporrà ancora domande di storia.**

1. **Packaging**

## [NFR9]:

L’applicazione dev’essere installata dal play-store di google oppure, alternativamente da un file di installazione apposito (. apk), che aggiungerà l’app tra le applicazioni già esistenti sul dispositivo.

# ****5.Ambiente di destinazione:****

**-Sistemi operativi che supportano l’app: Android**

**-Compatibilità: alta, supportata dal 70% dei dispositivi in circolazione.**

# ****6.Scadenze:****

6 novembre: Requisiti e casi d’uso

20 novembre: Requirements Analysis Document

3 dicembre: System Design Document

5 gennaio: piano di test e specifica interfacce dei moduli del sistema

29 gennaio: consegna progetto

# ****7.Criteri di accettazione dei test:****

**per completare la fase di testing, il sistema dovrà soddisfare i requisiti attesi e attenersi agli scenari elencati.**