



Università degli Studi di Padova



Catch em All - *CAPTCHA: Umano o Sovraumano?*

Email: [catchemallswe3@gmail.com](mailto:catchemallswe3@gmail.com)

# Analisi dei requisiti

Versione	(modifica)
Approvazione	(modifica)
Redazione	(modifica)
Verifica	(modifica)
Stato	(In sviluppo)
Uso	(modifica)
Distribuzione	(modifica)

## Registro delle modifiche

Versione	Descrizione	Data
0.0.2	Impostazione del layout	16/11/2022
0.0.1	Creazione e prime definizioni del documento	15/11/2022

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Scopo del documento . . . . .	3
1.2	Scopo del prodotto . . . . .	3
1.3	Glossario . . . . .	3
1.4	Riferimenti . . . . .	4
1.4.1	Riferimenti normativi . . . . .	4
1.4.2	Riferimenti informativi . . . . .	4
<b>2</b>	<b>Descrizione generale</b>	<b>5</b>
2.1	Caratteristiche del prodotto . . . . .	5
2.1.1	Task richiesta . . . . .	5
2.1.2	Dataset di immagini . . . . .	5
2.1.3	Algoritmo di rielaborazione di un immagine . . . . .	5
2.1.4	Algoritmo di controllo sulla correttezza della soluzione generata dall'utente . . . . .	5
2.1.5	Algoritmo di valutazione dei feedback . . . . .	6
2.2	Obblighi di Progettazione . . . . .	6
2.3	Requisiti opzionali . . . . .	6
2.4	Tecnologie utilizzate . . . . .	6
<b>3</b>	<b>Casi d'uso</b>	<b>7</b>
3.1	Obiettivi . . . . .	7
3.2	Attori . . . . .	7
3.3	UC1 - Autenticazione . . . . .	7
3.3.1	UC1.1 - Inserimento username . . . . .	8
3.3.2	UC1.2 - Inserimento password . . . . .	8
3.3.3	UC1.3 - Compilazione captcha . . . . .	8
3.3.4	UC1.3.1 - Compilazione captcha tipo 1 (chiedere quanto si può andare in dettaglio) . . . . .	8
3.3.5	UC1.3.2 - Compilazione captcha tipo 2 [in realtà non esiste, serve solo per dire che siamo bravi e abbiamo previsto la generalizzazione per una futura estensione del sistema, capire dove scrivere questa cosa] . . . . .	9
3.4	UC2 - Captcha non superato . . . . .	9
3.5	UC3 - Superamento del numero di tentativi consentito . . . . .	9
3.6	UC4 - Credenziali errate . . . . .	9
3.7	UC5 - Generazione di un altro captcha (da chiedere su una possibile generalizzazione) . . . . .	10
<b>4</b>	<b>Requisiti</b>	<b>11</b>
4.1	Requisiti funzionali . . . . .	11
4.2	Requisiti di qualità . . . . .	11
4.3	Requisiti di vincolo . . . . .	11
4.4	Requisiti prestazionali . . . . .	11
4.5	Tracciamento dei requisiti . . . . .	12
4.5.1	Requisito - Fonti . . . . .	12
4.5.2	Fonte - Requisiti . . . . .	12

4.6 Riepilogo dei requisiti . . . . .	13
---------------------------------------	----

# 1 Introduzione

## 1.1 Scopo del documento

In questo documento vengono forniti in modo esaustivo e completo i **requisiti** e i **casi d'uso** individuati dal gruppo Catch Em All a seguito dell'analisi approfondita del capitolato **CAPTCHA: Umano o Sovrumano?**

## 1.2 Scopo del prodotto

Dal proponente Zucchetti S.p.A. viene evidenziato, nel capitolato da loro proposto, una criticità negli attuali sistemi di sicurezza sulla rilevazione dei bot rispetto agli esseri umani. Oggi giorno il meccanismo più utilizzato per risolvere questo problema è il test CAPTCHA.

Un bot non è altro che una procedura automatizzata che, in questo caso, ha fini malevoli, come per esempio:

- registrazione presso siti web;
- creazione di spam;
- violare sistemi di sicurezza;

I bot, grazie alle nuove tecnologie sviluppate con sistemi che utilizzano principalmente l'intelligenza artificiale, riescono a svolgere compiti che fino a poco tempo fa venivano considerati impossibili da svolgere per una macchina.

Ciò evidenzia che i CAPTCHA attuali risultano sempre più obsoleti, non andando a individuare correttamente tutti i bot, se non quasi nessuno.

Un'altra criticità individuata dal proponente è il sistema di classificazione delle immagini che sta effettuando Google grazie al proprio reCAPTCHA, che attualmente è il sistema più diffuso.

Questa criticità nasce dal beneficio che questa big tech ottiene dall'interazione degli utenti nel risolvere le task proposte, che portano alla creazione di enormi dataset di immagini classificate che possono essere utilizzate per l'apprendimento dei propri sistemi di machine learning o vendibili a terzi.

Il capitolato C1 richiede di sviluppare una applicazione web costituita da una pagina di login provvista di questo sistema di rilevazione in grado di distinguere un utente umano da un bot.

L'utente quindi, dopo aver compilato il form in cui inserirà il nome utente e la password, dovrà svolgere una task che sarà il cosiddetto test CAPTCHA.

**(BISOGNERÀ POI DECIDERE IN FASE DI SVILUPPO SE PRIMA O DOPO)**

## 1.3 Glossario

Per evitare ambiguità relative al linguaggio utilizzato nei documenti prodotti, viene fornito il **Glossario v1.0.0** (per ora teorica la sua versione **DA MODIFICARE**). In questo documento sono contenuti tutti i termini tecnici, i quali avranno una definizione specifica per comprenderne al meglio il loro significato.

Tutti i termini inclusi, vengono segnalati all'interno del documento con una G a pedice. **(qui da vedere quando si implementa, lo scrivo perché tutti lo hanno scritto)**

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

Link vari da aggiungere in seguito

### 1.4.2 Riferimenti informativi

- Capitolato C1 “CAPTCHA: umano o sovrumano?” <https://www.math.unipd.it/~tullio/IS-1/2022/Progetto/C1.pdf>
- Slide T06 del corso di Ingegneria del Software – Analisi dei requisiti: <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T06.pdf>
- Slide P03 del corso di Ingegneria del Software – Diagrammi dei casi d’uso: <https://www.math.unipd.it/~rcardin/swea/2022/Diagrammi%20Use%20Case.pdf>;
- Regolamento del progetto didattico – Materiale didattico del corso di Ingegneria del Software: <https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/PD02.pdf>;

(TEORICAMENTE NON CE NE SONO ALTRI)

## 2 Descrizione generale

### 2.1 Caratteristiche del prodotto

L'obiettivo di questo progetto è il riconoscimento tra un essere umano ed un bot in maniera efficace, per offrire un'alternativa valida ai sistemi già esistenti che oramai non sono più in grado di svolgere questa task in maniera ottimale.

Di seguito vengono descritte le caratteristiche che dovrà avere il prodotto.

#### 2.1.1 Task richiesta

Il test CAPTCHA pensato richiederà all'utente di svolgere una task di classificazione. In particolare verrà fornito un insieme di 9 immagini, ognuna con una propria classe di appartenenza, con un totale di classi, presenti nell'insieme fornito, che può variare tra 2 ad un massimale di 4. All'utente, come task, verrà richiesto di selezionare le immagini appartenenti ad una particolare classe presente nell'insieme.

#### 2.1.2 Dataset di immagini

Verrà utilizzato il dataset pubblico Unsplash, un dataset di immagini senza copyright, il quale fornisce delle API che permettono il prelevamento di una immagine data una parola chiave, che corrisponderà poi alla classe di appartenenza nel nostro database.

#### 2.1.3 Algoritmo di rielaborazione di un immagine

Il team si occuperà di generare un algoritmo ad hoc, in linguaggio Python, che presa un'immagine normale dal dataset Unsplash, individua il soggetto dell'immagine per poi ridefinirne solo i contorni in bianco e lasciando tutto il resto dell'immagine in nero.

#### 2.1.4 Algoritmo di controllo sulla correttezza della soluzione generata dall'utente

Ricevuta la soluzione generata dall'utente in risposta al CAPTCHA fornitogli, vi è la necessità di creare un algoritmo che verifichi la correttezza o meno della soluzione proposta. In caso di correttezza, l'utente effettuerà il login, altrimenti gli verrà generato un altro CAPTCHA da risolvere, con immagini diverse e un'altra classe da individuare all'interno dell'insieme, diminuendo di un numero il numero di tentativi rimasti per autenticarsi. Al termine di questi tentativi, l'utente verrà bloccato per 20 minuti prima di poter riprovare ad accedere. **(Qui ho proprio sparato la prima soluzione pensata, si deciderà in futuro e si modificherà qui di conseguenza).**

Inoltre questo algoritmo se e solo se **(HO RAGIONATO SUL PERCHÉ NON TENERE CONTO DELLE SOLUZIONI SBAGLIATE, IN CASO VI SPIEGO IO PERCHÉ NON FUNZIONA ANALIZZARE ANCHE QUELLE SBAGLIATE)** la soluzione proposta dall'utente è corretta, richiamerà un algoritmo di valutazione del feedback ricevuto, descritto nel prossimo punto.

### 2.1.5 Algoritmo di valutazione dei feedback

Un obiettivo fondamentale che si è posto il team è di creare un sistema automatizzato che sia scalabile per ovviare al problema di utilizzare un numero fisso e limitato di immagini nei CAPTCHA generati alle diverse richieste. Come già descritto, per ogni CAPTCHA verranno utilizzate delle immagini, in parte, su cui non si ha un'affidabilità elevata che saranno generate dal sistema sopra descritto. Queste immagini in base all'interazione degli utenti, verranno categorizzate affidabili o non, al fine di ingrandire man mano il dataset di immagini affidabili da utilizzare.

## 2.2 Obblighi di Progettazione

- Sviluppare una applicazione web costituita da una pagina di login che presenti un sistema in grado di distinguere un utente umano da un robot;
- Verifica che dimostri che il sistema CAPTCHA non è eludibile chiamando in modo diretto la componente server senza aver utilizzato la parte client;
- Analisi sulle tecnologie utilizzate, al fine di indicare quali sviluppi futuri di diverse tecnologie possono con il tempo rendere inefficace il sistema di verifica;
- Il sistema di CAPTCHA potrà essere una libreria Open Source, un servizio obbligatoriamente gratuito fruibile via web.

## 2.3 Requisiti opzionali

- Form di registrazione di un nuovo utente;
- Mini-forum che accetta contenuti prodotti dagli utenti dell'applicazione;
- Pagina di ricerca sul forum con verifica CAPTCHA;

## 2.4 Tecnologie utilizzate

Per sviluppare la piattaforma verranno utilizzate le seguenti tecnologie:

**DA SCRIVERE DURANTE LO SVILUPPO**



## 3 Casi d'uso

### 3.1 Obiettivi

La sezione 3 Casi d'uso ha come obiettivo l'identificazione e la descrizione di tutti i casi d'uso, ovvero interazioni tra sistema ed attori, individuati dagli analisti nel tempo tramite lo studio del capitolato d'appalto, del dominio, e tramite incontri con il committente.

### 3.2 Attori

Dato che il requisito obbligatorio richiede la costruzione di una pagina di login(probabilmente username e password saranno preimpostati) che presenti un sistema in grado di distinguere un utente umano da un robot, il prodotto presenterà una sola tipologia di utente:



L'utente generico, che potrà essere una persona fisica o anche un bot, potrà accedere a tutte le funzionalità del prodotto.

TODO aggiungere eventualmente come utenti il db di unsplash e servizi vari in futuro

### 3.3 UC1 - Autenticazione

**Attore primario:** Utente generico

**Precondizioni:** Il sistema non riconosce l'utente

**Postcondizioni:** L'utente è autenticato nel sistema

**Scenario principale:** L'utente: Inserisce il proprio username [UC1.1]

Inserisce la propria password [UC1.2]

Compila correttamente il captcha visualizzato [UC1.3]

**Scenari alternativi:**

1. L'utente inserisce delle credenziali errate, il sistema mostra un errore e fa ritentare l'autenticazione all'utente. [UC4]
2. L'utente non supera il captcha [uc"2"]: //da modificare UC3 diventa estensione di UC2
  - (a) Nel caso in cui l'utente non abbia raggiunto il numero di tentativi massimo [UC2]:
    - i. Viene visualizzato un errore esplicativo
    - ii. L'utente non viene autenticato nel sistema
    - iii. Il sistema fa ritentare l'autenticazione all'utente
  - (b) Nel caso in cui l'utente abbia superato il numero di tentativi consentito per il superamento del captcha negli ultimi 20 minuti [UC3]:
    - i. L'utente visualizza un messaggio di blocco
    - ii. L'utente non viene autenticato nel sistema

- iii. Il sistema bloccherà futuri tentativi di autenticazione da parte di quell'utente per un tempo prestabilito

### 3.3.1 UC1.1 - Inserimento username

**Attore primario:** Utente generico

**Precondizioni:** Il sistema non riconosce l'utente e non conosce l'username

**Postcondizioni:** Il sistema ha ricevuto l'username dell'utente

**Scenario principale:**

1. L'utente inserisce l'username nell'apposito spazio. L'username deve essere quello preimpostato.

### 3.3.2 UC1.2 - Inserimento password

**Attore primario:** Utente generico

**Precondizioni:** Il sistema non riconosce l'utente e non conosce la password

**Postcondizioni:** Il sistema ha ricevuto la password dell'utente

**Scenario principale:**

1. L'utente inserisce la password nell'apposito spazio. L'username deve essere quello preimpostato.

### 3.3.3 UC1.3 - Compilazione captcha

**Attore primario:** Utente generico

**Precondizioni:** L'utente visualizza il captcha proposto dal sistema

**Postcondizioni:** L'utente ha risolto il captcha (non necessariamente in modo corretto)

**Scenario principale:**

1. L'utente risolve il captcha proposto, non necessariamente in maniera corretta

**Generalizzazioni:**

- captcha tipo 1 [UC1.3.1]
- captcha tipo 2 [UC1.3.2]
- ...

### 3.3.4 UC1.3.1 - Compilazione captcha tipo 1 (chiedere quanto si può andare in dettaglio)

**Attore primario:** utente generico

**Precondizioni:** l'utente visualizza il captcha tipo 1

**Postcondizioni:** l'utente ha completato il captcha

**Scenario principale:**

All'utente viene presentato il captcha tipo 1, con 9 immagini da classificare L'utente classifica le 9 immagini nelle giuste categorie

### **3.3.5 UC1.3.2 - Compilazione captcha tipo 2 [in realtà non esiste, serve solo per dire che siamo bravi e abbiamo previsto la generalizzazione per una futura estensione del sistema, capire dove scrivere questa cosa]**

**Attore primario:** utente generico

**Precondizioni:** l'utente visualizza il captcha tipo 2

**Postcondizioni:** l'utente ha completato il captcha

**Scenario principale:**

All'utente viene presentato il captcha tipo 2 [elenco delle azioni che servono per superare il captcha]

## **3.4 UC2 - Captcha non superato**

**Attore primario:** Utente generico

**Precondizioni:** L'utente non ha compilato il captcha correttamente

**Postcondizioni:** L'utente visualizza un messaggio di errore e l'operazione di autenticazione fallisce. L'utente potrà riprovare ad autenticarsi

**Scenario principale:**

1. L'utente visualizza un messaggio di errore
2. Il numero di tentativi compiuti dall'utente negli ultimi 20 minuti aumenta di 1
3. L'operazione di autenticazione fallisce
4. Viene generato un altro captcha e l'utente potrà riprovare ad autenticarsi

## **3.5 UC3 - Superamento del numero di tentativi consentito**

**Attore primario:** Utente generico

**Precondizioni:** L'utente ha superato il numero massimo di tentativi di autenticazione consentiti e non ha compilato l'ultimo captcha correttamente

**Postcondizioni:** L'operazione di autenticazione fallisce e l'utente viene bloccato

**Scenario principale:**

1. L'utente visualizza un messaggio di errore che comunica lo stato di blocco
2. L'utente conferma di aver visualizzato il messaggio di errore
3. L'operazione di autenticazione fallisce
4. L'utente viene bloccato da futuri tentativi di login per un tempo prestabilito

## **3.6 UC4 - Credenziali errate**

**Attore primario:** Utente generico

**Precondizioni:** L'utente non ha inserito le credenziali corrette

**Postcondizioni:** L'utente visualizza un messaggio di errore e l'operazione di autenticazione fallisce. L'utente potrà riprovare ad autenticarsi

**Scenario principale:**

1. L'utente visualizza un messaggio di errore
2. L'operazione di autenticazione fallisce
3. Viene generato un altro captcha e l'utente potrà riprovare ad autenticarsi

### **3.7 UC5 - Generazione di un altro captcha (da chiedere su una possibile generalizzazione)**

**Attore primario:** Utente generico

**Precondizioni:** L'utente ha visualizzato il captcha proposto

**Postcondizioni:** All'utente viene proposto un nuovo captcha

**Scenario principale:**

1. L'utente richiede la generazione di un nuovo captcha
2. Il sistema propone all'utente un altro captcha

## 4 Requisiti

In questa sezione viene riportato l'elenco completo dei requisiti del progetto. I requisiti sono stati individuati e classificati durante l'attività di analisi dei requisiti, e a ciascuno è stato assegnato un codice identificativo univoco secondo quanto stabilito nelle Norme di progetto.

### 4.1 Requisiti funzionali

ID Requisito	Classificazione	Descrizione
RF-1	Obbligatorio	L'utente deve poter raggiungere il sistema in
RF-2	Obbligatorio	L'utente deve potersi autenticare all'interno del sistema tran
RF-3	Obbligatorio	L'utente deve poter inserire il proprio usern
RF-4	Obbligatorio	L'utente deve poter inserire la propria passw
RF-5	Obbligatorio	L'utente deve superare il captcha proposto dal sistema per po
RF-6	Obbligatorio	L'utente deve superare il captcha tipo 1 per poter effe
RF-7	Obbligatorio	All'utente viene mostrato un messaggio di errore nel caso in cui non
RF-8	Obbligatorio	All'utente viene mostrato un messaggio di errore nel caso in cui abbia superato il
RF-9	Obbligatorio	L'utente deve poter richiedere la generazione di un a
RF-10	Obbligatorio	All'utente viene mostrato un messaggio di errore nel caso in cui abbia ins

### 4.2 Requisiti di qualità

ID Requisito	Classificazione	Descrizione
RQ-1	Obbligatorio	Il captcha deve mitigare potenziali attacchi "brute force"
RQ-2	Obbligatorio	Il server non deve accettare chiamate con dati falsificati
RQ-3	Obbligatorio	Dovrà essere prodotto un manuale utente del prodotto
RQ-4	Obbligatorio	Dovrà essere prodotto un manuale utente per l'estensione del prodotto
RQ-5	Obbligatorio	Il prodotto dovrà essere sviluppato rispettando quanto espresso nelle Norme di p
RQ-6	Obbligatorio	La documentazione e il codice sorgente del prodotto saranno pubblicate su Git
RQ-7	Obbligatorio	Dovrà essere condotta un'analisi sulle tecnologie utilizzate
RQ-8	Obbligatorio	Il sistema dovrà essere distribuito con la licenza ???

### 4.3 Requisiti di vincolo

ID Requisito	Classificazione	Descrizione
RV-1	Obbligatorio	Il sistema deve essere raggiungibile in cloud
RV-2	Obbligatorio	Il sistema deve funzionare con la versione più recente del browser Chrome
RV-3	Obbligatorio	Le librerie utilizzate devono essere open source
RV-4	Obbligatorio	I servizi web utilizzati devono essere gratuiti
RV-5	Obbligatorio	La parte server deve essere sviluppata utilizzando il linguaggio Python
RV-6	Obbligatorio	La parte client deve essere sviluppata utilizzando i linguaggi HTML, CSS, JavaS

### 4.4 Requisiti prestazionali

Non sono stati individuati requisiti prestazionali misurabili.

## 4.5 Tracciamento dei requisiti

### 4.5.1 Requisito - Fonti

Requisito	Fonti
RF-1	Capitolato
RF-2	UC1, Capitolato
RF-3	UC1.1, Capitolato
RF-4	UC1.2, Capitolato
RF-5	UC1.3, Capitolato
RF-6	UC1.3.1, VE-20221111
RF-7	UC2
RF-8	UC3, VI-20221115
RF-9	UC4, VI-20221115
RF-10	UC5
RQ-1	Capitolato
RQ-2	Capitolato
RQ-3	Capitolato
RQ-4	Capitolato
RQ-5	VI-20221115
RQ-6	VI-20221115
RQ-7	Capitolato
RQ-8	VI-20221115
RV-1	Capitolato
RV-2	VI-20221115
RV-3	Capitolato
RV-4	Capitolato
RV-5	VE-20221021
RV-6	Capitolato

### 4.5.2 Fonte - Requisiti

Fonte	Requisiti
UC1	RF-2
UC1.1	RF-3
UC1.2	RF-4
UC1.3	RF-5
UC1.3.1	RF-6
UC2	RF-7
UC3	RF-8
UC4	RF-9
UC5	RF-10
Capitolato	RF-1, RF-2, RF-3, RF-4, RF-5, RQ-1, RQ-2, RQ-3, RQ-4, RQ-7, RV-1, RV-3, RV-4, RV-6
VE-20221111	RF-6
VE-20221021	RV-5
VI-20221115	RF-8, RF-9, RQ-5, RQ-6, RQ-8, RV-2

#### 4.6 Riepilogo dei requisiti

Classificazione	Obbligatorio	Facoltativo	Totale
Funzionali	10	0	10
Di qualità	8	0	8
Di vincolo	5	1	6
<b>Totale</b>	<b>23</b>	<b>1</b>	<b>24</b>