

Corso di Laurea in Informatica, prof. A. De Lucia, a.a. 2021-22

Progetto di Ingegneria del Software

Immagine che contiene testo

Descrizione generata automaticamente  
  
[» GitHub Repository](https://github.com/is-shodan-21-22)

Scadenza: 07-10-2021❯ Project Proposal

Scadenza: 14-10-2021❯ Problem Statement

Scadenza: 28-10-2021❯ Requisiti e casi d’uso

Scadenza: 11-11-2021❯ Requisiti e casi d’uso

Scadenza: 25-11-2021  
➥ **System Design Document**

Scadenza: 16-12-2021❯ Piano di test e specifica interfacce moduli

Scadenza: alla consegna del progetto❯ Altri documenti

❯ Esecuzione dei test

❯ Object Design

❯ Codice sorgente

❯ Report sul lavoro individuale

**Shodan** [IS-2021/2022-DE LUCIA]  
Partecipanti del progetto

|  |  |  |
| --- | --- | --- |
| Nome | Matricola | E-mail |
| Antonio Gravino | 05121 07161 | a.gravino3@studenti.unisa.it |
| Dario Trinchese | 05121 07479 | d.trinchese2@studenti.unisa.it |
| Raffaele Zheng | 05121 09015 | r.zheng@studenti.unisa.it |
| Carmine Fabbri | 05121 07353 | c.fabbri@studenti.unisa.it |
| Carmine Napolitano | 05121 06417 | c.napolitano44@studenti.unisa.it |

Questo documento tratta esclusivamente il **SDD** del progetto.  
Per ulteriore documentazione valida ai fini dell’esame, consultare la repository *docs*.

**1.0 – Introduzione**

**1.1 – Scopo del sistema**

**Shodan è una piattaforma per il *retro-gaming***, ed è dedicata in particolar modo ai fan sfegatati di alcuni franchise videoludici. Shodan, che prende il proprio nome da un personaggio di un titolo pubblicato negli anni 90, si propone di diventare un punto di riferimento per tutti quei videogiocatori che, per amor dell’industria videoludica contemporanea, intendono e desiderano dare uno sguardo al passato dei videogiochi. Tramite Shodan si potranno **acquistare videogiochi rilasciati prima del 2000** (e quindi con oltre venti anni d’età, da qui “*retro*” in “*retro-gaming*”), sfruttare saldi per accaparrarsi le migliore offerte e **collezionare titoli nella propria libreria digitale.**

**1.2 – Obiettivi del sistema**

Il sistema “Shodan” è stato progettato considerando i seguenti obbiettivi di design:

**1.2.1 – Criteri di performance:**

• Tempo di risposta: Il sistema deve essere reattivo e in grado di servire più utenti contemporaneamente.

• Usabilità: Il sistema deve essere accessibile a quanti più utenti possibile, in modo da poter permettere a chiunque di poter inserire o visionare prodotti in maniera semplice e guidata da qualunque dispositivo utilizzato.

**1.2.2** – **Criteri di affidabilità:**

• Sicurezza: Il sistema deve garantire la protezione dei dati sensibili scambiati con l’utente ed evitare che terzi vi accedano, attraverso anche paradigmi di programmazione crittografati.

• Disponibilità: Il sistema deve essere disponibile ad un alto numero di utenti.

• Robustezza: Il sistema deve essere in grado di gestire correttamente l’immissione di eventuali input errati.

**1.2.3** **– Criteri di manutenzione:**

• Modificabilità: Il sistema deve poter essere facilmente modificabile in modo da correggere eventuali errori.

• Resistenza agli errori: Il sistema deve gestire in modo corretto eventuali eccezioni software.

**1.3 – Definizione, acronimi e abbreviazioni**

RAD = Requirement Analysis Document

SDD = System Design Document

MVC = Model View Control

JSP = Java Servlet Page

**1.4 – Riferimenti**

Riferimento al Requirement Analysis Document di Shodan:

**1.5 – Panoramica**

Il seguente documento di System Design (SDD) mostra i dettagli tecnici del design del sistema Shodan. Altri dettagli riguardanti le funzionalità e le caratteristiche del sistema possono essere trovati nel documento dell'analisi dei requisiti (RAD) mentre una panoramica generale può essere trovata nel Problem Statement. In tale documento viene specificata una introduzione generale all'architettura e agli obiettivi di design che il sistema si propone di raggiungere, viene proposta una suddivisione del sistema in sottosistemi definendone inoltre il mapping Hardware/Software, in modo da assegnare ogni sottosistema ad uno specifico hardware. Sono inoltre descritti il controllo dell'accesso e i problemi di sicurezza legati al sistema con relativi paradigmi di crittografia utilizzati, evidenziando anche il controllo generale del software e il controllo dei Boundary, trattando gli stati iniziali e la gestione del sistema.

**2.0 – Architettura Software Attuale**

L’architettura software attuale non è esistente

**3.0 – Architettura Software Proposta**

**3.1 – Panoramica**

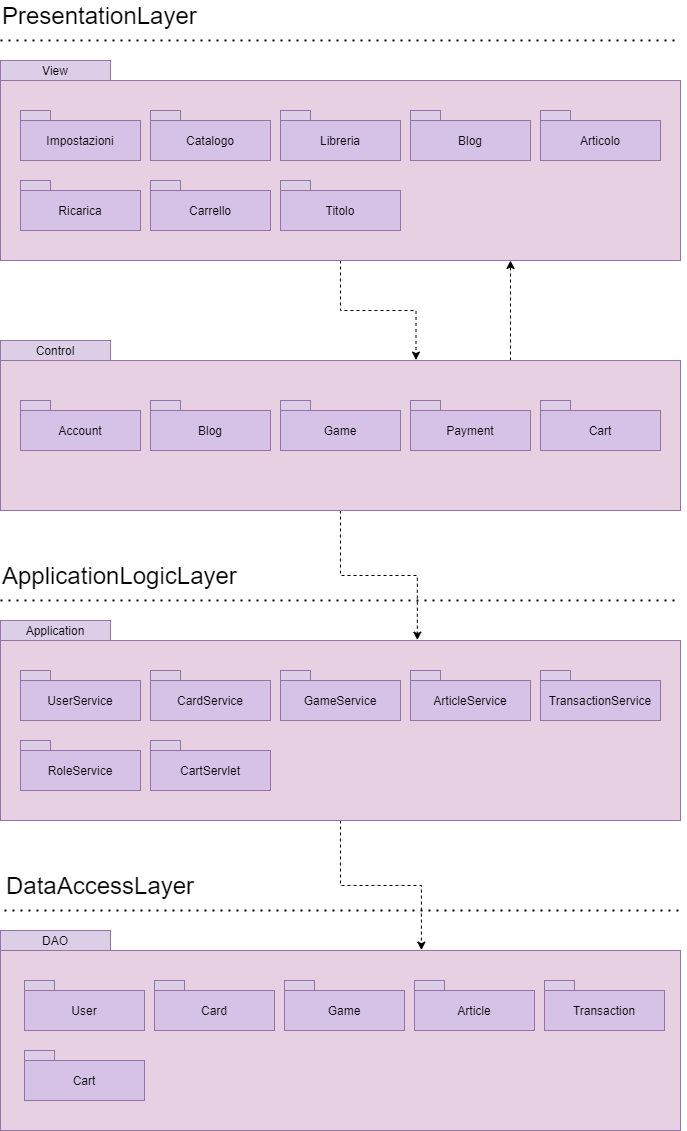
Il nostro sistema adotta l’architettura MVC (Model View Control), che generalmente viene applicata ai sistemi web. I sottosistemi principali, previsti dalla stessa architettura MVC sono:

**Model**: rappresenta il sistema di gestione dei dati. Si occupa della memorizzazione dei dati, come l’interazione con i database.

**View**: rappresenta il sistema di interazione diretta con l’utente; rappresenta in tutto e per tutto l’interfacciamento che il sistema ha con tutti gli utenti che possono interagire con il sistema.

**Controller**: in questo sottosistema sono presenti le componenti che utilizzano ed elaborano i dati; rappresenta il cuore del sistema.

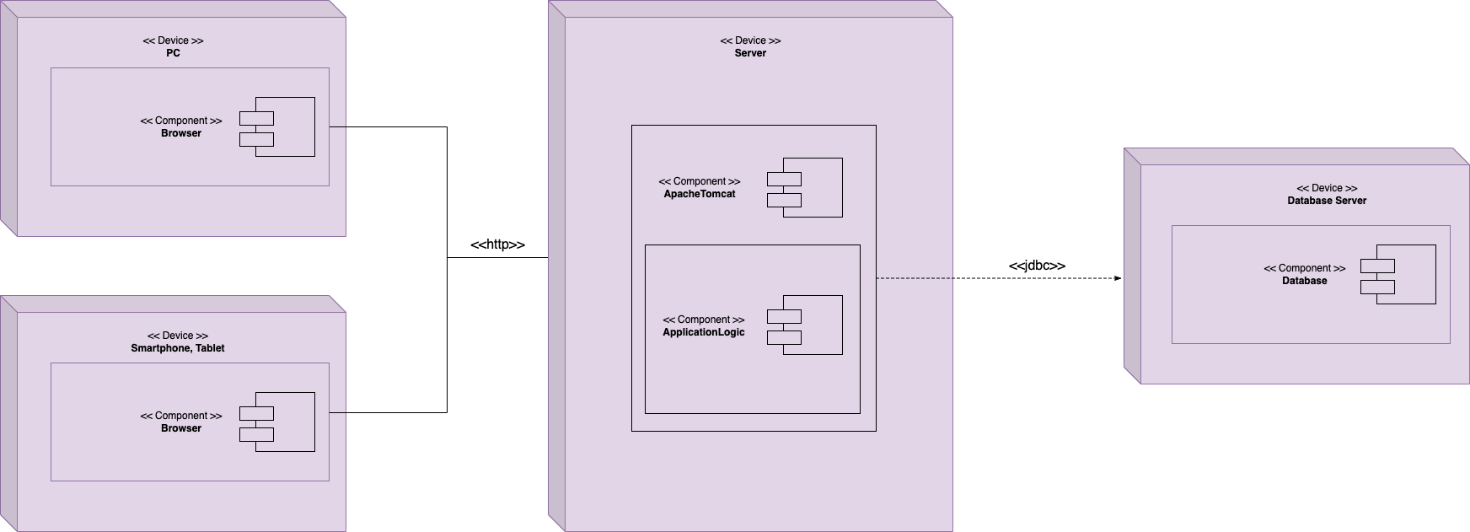
**3.2 – Decomposizione in sottosistemi**



|  |  |
| --- | --- |
| **Gestione Account** | Comprende le funzionalità di “Login”, “Logout” e “Registrazione”, Visualizza Informazioni personali”,  “Modifica e-mail”, “Modifica password” |
| **Gestione Blog** | Comprende le funzionalità “Aggiungi notizia al blog”, “Rimuovi notizia dal blog”, “Visualizza blog”, “Leggi notizia” |
| **Gestione Game** | Comprende le funzionalità “Aggiungi titolo al catalogo”, “Rimuovi titolo dal catalogo”, “Modifica titolo nel catalogo”, “Visualizza catalogo”, “Visualizza titolo”, “Ricerca titolo” |
| **Gestione Payment** | Comprende le funzionalità “Ricarica saldo”, “Aggiungi carta”, “Paga ora” |
| **Gestione Cart** | Comprende le funzionalità “Aggiungi titolo”, “Rimozione titolo”, “Svuota carrello”,” Visualizza il carrello” |

**4.0 – Mappatura Hardware/Software**

Il sistema utilizza un’architettura Client/Server. Il Web Server è rappresentato da Apache Tomcat 9 ed è situato su una singola macchina, la logica del sistema è costituita da Java Servlet mentre l’interfaccia utente è realizzata utilizzando pagine JSP (Java Servlet Page). Il Client è rappresentato dal Web Browser utilizzato dall’utente. La comunicazione tra i nodi è rappresentata da richieste e risposte http tra client e server, e da query in JDBC tra server e database.



**7.0 – Controllo Software Globale**

Essendo "**Shodan**" un’applicazione web, il Web Server si occupa di gestire le varie richieste dei client. Il server smista le richieste alle classi Java Servlet opportune che si occuperanno di gestire la richiesta, eventualmente interagire con il model, e dare una risposta. Dopodiché il server crea la pagina JSP che verrà poi convertita in pagina html e visualizzata dall’utente.

**8.0 – Boundary Conditions**

Per consentire il corretto accesso al sistema, il cliente necessita di una connessione ad Internet e di un browser moderno, in modo che possa visualizzare il sito correttamente. Una volta entrati sul sito, non sono necessarie altre configurazioni per usufruire del sistema.

* 1. **Scenari di avvio**

|  |  |
| --- | --- |
| **Nome scenario** | Startup del Sistema |
| **Attori partecipanti** | Antonio: Amministratore del Sistema |
| **Flusso di eventi** | 1. Antonio accede al sistema operativo per avviare i servizi del web server e del database.  2. Antonio avvia il pannello di controllo e clicca sulla voce “Servizi”  3. In questa schermata viene mostrata una lista dei servizi e il loro stato (che può essere in esecuzione o non avviato).  4. Antonio individua il servizio relativo al database MySQL, lo seleziona e premendo il tasto destro su di esso, appare una lista di opzioni. Qui Antonio clicca sulla voce “Avvia”.  5. Antonio successivamente individua il servizio relativo al web server Tomcat, lo seleziona e preme il tasto destro su di esso, appare una lista di opzioni. Qui clicca sulla voce “Avvia”.  6. A questo punto il sistema è totalmente avviato. |

|  |  |
| --- | --- |
| **Nome scenario** | Shutdown del Sistema |
| **Attori partecipanti** | Antonio: Amministratore del Sistema |
| **Flusso di eventi** | 1. Antonio accede al sistema operativo per fermare i servizi del web server e del database.  2. Antonio avvia il pannello di controllo e clicca sulla voce “Servizi”  3. In questa schermata viene mostrata una lista dei servizi e il loro stato (che può essere in esecuzione o non avviato).  4. Antonio individua il servizio relativo al web server Tomcat, lo seleziona e preme il tasto destro su di esso, appare una lista di opzioni. Qui clicca sulla voce “Ferma”.  5. Antonio successivamente individua il servizio relativo al database MySQL, lo seleziona e premendo il tasto destro su di esso, appare una lista di opzioni. Qui clicca sulla voce “Ferma”.  6. A questo punto il sistema è spento. |

* 1. **Fallimenti**

Possono verificarsi diversi casi di fallimento del sistema:

• Nel caso in cui vi fosse l’interruzione della connettività con la rete, il servizio resterà interrotto fino a quando il collegamento non sarà ripristinato.

• Nel caso di guasto al disco dove risiede il database, è prevista la creazione periodica di backup in modo da ripristinare lo stato

**8.3 Tools per sviluppatori**

Per gli sviluppatori sarà messo a disposizione un semplice installer, la cui esecuzione solleciterà, in sequenza, il download dei repositories di Shodan da GitHub.com, l’avvio automatico dei servizi di Database Server MySQL e Web Server Tomcat, il deployment dell’applicazione web “Shodan” sul Web Server e l’apertura della repository di sviluppo nell’IDE preferito.