Gestione casinò

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 6

1.8 Analisi dei mezzi 6

1.8.1 Software 6

1.8.2 Hardware 6

2 Progettazione 7

2.1 Design dell’architettura del sistema 7

2.2 Design dei dati e database 7

2.3 Design delle interfacce 7

2.4 Design procedurale 7

3 Implementazione 8

4 Test 8

4.1 Protocollo di test 8

4.2 Risultati test 9

4.3 Mancanze/limitazioni conosciute 9

5 Consuntivo 9

6 Conclusioni 9

6.1 Sviluppi futuri 9

6.2 Considerazioni personali 9

7 Bibliografia 9

7.1 Bibliografia per articoli di riviste: 9

7.2 Bibliografia per libri 9

7.3 Sitografia 9

8 Allegati 10

# Introduzione

## Informazioni sul progetto

Autore: Matan Davidi, Thor Düblin, Matteo Forni, Carlo Pezzotti, Mattia Toscanelli

Scuola: Arti e Mestieri Trevano

Classe: I3AA

Anno scolastico: 2019

Sezione: Informatica

Materia: Modulo 306

Docenti responsabili: Massimo Sartori

Data di inizio: 13.02.2019

Data di consegna: 17.05.2019

## Abstract

This document contains the documentation of the realization of a casino’s management software. This software has to manage the casino’s users, its games, its rooms and the promotions it can offer to its users. Also, games and rooms can have images or videos to accompany them.

The software’s realization has to implement a system of continuous integration and automatic testing that works alongside a version control tool, such as GitHub, so that it doesn’t allow the team that is working on it to push something to the repository without first checking that it passes every test for quality control. Basically, this way, nothing that breaks the functioning of the application can be pushed to production.

Also, a user’s name, surname, address, house number, zip code, city, email address, phone number and gender registered in the system’s database. Each game has a name and a room where it is played, each room a location where it is and, for each media file, the URL, game or room it represents, and its type (picture, video, …) are stored. Finally, the promotions are handled by storing the message they show and the user they are shown to.

The quality controls consist in three subcategories:

* Emails (checking if the system is able to send an email)
* Database (testing if the system can connect to the database, testing if the system cannot connect with invalid credentials, testing if the system is able to insert a new user, testing if the system cannot insert a new user with an invalid format, testing if the results returned from a query are correct, testing if the system does not return information requested by a malformed query)
* Users (controlling that a new user can be created before inserting it into the database, controlling that a new user cannot be created if its data is invalid, controlling if each value inserted for a new user is valid and not invalid)

## Scopo

Lo scopo di questo progetto è quello di creare un’applicazione web che semplifichi la gestione di un casinò in ogni suo aspetto: gli utenti, i giochi, le sale e le promozioni.

Inoltre un altro obiettivo è quello di insegnarci a utilizzare sistemi di test e di integrazione continua che vengono usati anche in grandi aziende come Selenium e Jenkins in modo da poterli utilizzare in futuro quando ci ritroveremo a lavorare in una ditta vera.

## Analisi

## Analisi del dominio

Il dominio per questa applicazione è pressoché inesistente, in quanto l’applicazione per la gestione del casinò deve venir fatta da capo partendo dai requisiti del cliente e non esistono applicazioni che potremmo usare come modello dalle quali ispirarci.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| Nome | Sito web per la gestione di un casinò |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| Nome | Gestione degli utenti |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |
| *Sotto-requisiti* | |
| 001 | Per ogni utente devono essere memorizzati il nome, cognome, l’indirizzo, il numero civico, la città, il NAP, l’indirizzo email, il numero di telefono, il sesso e una password. |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| Nome | Gestione giochi |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |
| *Sotto-requisiti* | |
| 001 | Per ogni gioco bisogna gestire il nome e la sala nella quale si trova. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| Nome | Gestione sale |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |
| *Sotto-requisiti* | |
| 001 | Per ogni sala deve essere gestita la posizione all’interno casinò. |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| Nome | Gestione promozioni |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |
| *Sotto-requisiti* | |
| 001 | Devono essere gestite delle promozioni da mostrare a uno o più utenti sotto forma di messaggi pop-up. |
| 002 | Ogni promozione deve contenere un messaggio da mostrare a un utente |
| 003 | Deve essere gestito quale promozione viene visualizzata da quali utenti |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| Nome | Protocolli di test e integrazione continua |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |
| *Sotto-requisiti* | |
| 001 | Devono essere implementati dei protocolli di test che permettano di verificare il funzionamento totale dell’applicazione. |
| 002 | I protocolli di test devono essere avviati ogni volta che viene eseguito un push verso il repository di GitHub. |
| 003 | I protocolli di test comprendono la verifica che le email vengano inviate correttamente |
| 004 | I protocolli di test comprendono la verifica della corretta connessione al database |
| 005 | I protocolli di test comprendono la verifica della validità dei dati inseriti dall’utente in fase di registrazione |

## Pianificazione



Figura : diagramm di Gantt utilizzato per pianificare il progetto

La pianificazione si divide in 5 fasi distinte: Stesura del Gantt, Analisi, Progettazione, Implementazione, Protocolli di test; ognuna delle quali si suddivide nuovamente in attività.

### Stesura del Gantt



Figura : Attività "Stesura del Gantt"

La stesura del Gantt è quell’attività che porta ad avere una pianificazione del progetto e consiste in due operazioni:

* Discutere su eventuali attività, che consiste nel discutere tra i componenti del gruppo i passi necessari per riuscire a realizzare il progetto.
* Riportare le attività, che implica creare la pianificazione del progetto grazie a uno strumento come GanttProject.

### Analisi



Figura : Attività "Analisi"

L’analisi del progetto in questo caso è consistita nell’informarsi ognuno sui nuovi software che si sarebbe ritrovato a utilizzare. Questo significa che Matteo Forni si è informato sull’installazione e l’utilizzo di Jenkins, Carlo sull’utilizzo di PHPUnit, Thor sull’utilizzo di Selenium e JUnit. Inoltre abbiamo effettuato le seguenti operazioni:

* Analisi del dominio, ovvero l’analisi della situazione attuale prima della realizzazione del progetto che permette di valutare se ha senso realizzare il progetto.
* Analisi dei requisiti, che consiste nell’analizzare le richieste del cliente e stilare una lista di requisiti che l’applicazione deve soddisfare prima di poter essere consegnata al committente.
* Analisi dei costi, che implica un’analisi di costi e benefici del progetto in modo da definire se vale la pena dal nostro punto di vista creare l’applicazione o se i costi sono maggiori dei benefici.

### Progettazione

****

Figura : Attività "Progettazione"

La progettazione è consistita nell’organizzare la sua implementazione in modo da dividere il lavoro all’interno del team e non avere problemi all’interno del team in cui due componenti stanno lavorando su cose contrastanti.

### Implementazione



Figura : Attività "Implementazione"

L’implementazione è stata la parte più lunga del progetto oltre alla documentazione. È l’attività durante la quale abbiamo dovuto realizzare il progetto in ogni suo aspetto, basandoci sulla progettazione fatta nel punto precedente. Essa è divisa nelle seguenti operazioni:

* Installare ambiente, che a sua volta è categorizzato da:
  + Installare server, ovvero l’installazione e la configurazione di tutti i componenti di base necessari per l’utilizzo del sistema operativo
* Configurare ambiente, che si divide in:
  + Installare Java
  + Installare MySQL
  + Installare PHP
  + Installare Jenkins
    - Configurare Jenkins, per farlo funzionare in modo che ad ogni push sul repository di GitHub vengano eseguiti i protocolli di test che verifichino se il codice funziona anche dopo il push
  + Installare Selenium
    - Configurare Selenium, per fargli eseguire i test dell’interfaccia grafica su “ordine” di Jenkins dopo ogni push
  + Installare PHPUnit
* Realizzare Frontend, ovvero realizzare le seguenti pagine web:
  + Pagina di benvenuto
  + Pagina di registrazione
  + Pagina di login
  + Pagina sala giochi
* Realizzare Backend, ossia la creazione del codice che permette di eseguire tramite interfaccia web delle operazioni. Più nello specifico:
  + Creare database fisico, la realizzazione del database MySQL sul quale si basa l’intera applicazione
  + Gestione utenti
    - Creare/modificare/eliminare utente
  + Gestione registrazioni
    - Controllo creazione
* Protocolli di test
  + Test grafici Selenium
    - Test link pagina di benvenuto
      * Test link sala giochi
      * Test link login
    - Test login
      * Registrare un nuovo utente
      * Accedere con utente
  + Test logici Jenkins
    - Inserire/Modificare/Eliminare dati database
    - Verifica login con utente
    - Validazione dei dati
    - Creare/modificare/eliminare utenti
    - Verificare condizioni di creazione utenti

### Protocolli di test



Figura : Attività "Protocolli di test"

### Documentazione di progetto

Infine c’è un’attività rimasta costante durante tutto il progetto: la documentazione. Infatti chi aveva dei tempi morti da riempire lo faceva documentando il progetto e aggiungendo informazioni al file che state leggendo in questo momento in modo da non doverla scrivere tutta in poco tempo alla fine.

## Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

### Software

Programmi installati:

* PuTTY versione 0.70

Librerie di codice utilizzate:

* Bootstrap 4.3.1
* PHPMailer 6.0.7
* jQuery 3.3.1
* Notify.js 2015
* JUnit Jupiter 5.0-M1

### Hardware

Macchina server:

* Ubuntu Server 18.4
* 1 GB di memoria RAM
* 25 GB di disco disponibili

# Progettazione

Questo capitolo descrive esaustivamente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all’esecutore di evitare fraintendimenti e imprecisioni nell’implementazione del prodotto.

## Design dell’architettura del sistema

Descrive:

* La struttura del programma/sistema lo schema di rete...
* Gli oggetti/moduli/componenti che lo compongono.
* I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
* Eventuale sitemap

## Design dei dati e database

Il database sviluppato per essere utilizzato con questa applicazione è stato progettato tramite i seguenti schema E/R e logico:



Come prima cosa, a sinistra troviamo la tabella dell’**Utente**, che viene identificato da nome, cognome, sesso (i cui valori predefiniti sono “Male” e “Female” contenuti nella tabella **Gender**), via, numero civico, NAP, città, indirizzo e-mail, numero di telefono, password e un valore booleano che dice se è stato verificato o meno. In seguito abbiamo la tabella **Gioco**, che contiene un nome e un riferimento a una sala. Poi troviamo **Sala**, identificata unicamente da un luogo, ossia un valore predefinito contenuto nella tabella **Location**. Andando avanti c’è **Media**, ossia la tabella che contiene i riferimenti alle immagini e i video associati ai giochi o alle sale, che contiene l’URL che punta a quest’immagine o video, il gioco o sala a cui è associato e il tipo di file multimediale di cui si tratta, appunto video, immagine, … Questi sono valori specificati all’interno della tabella **Type**. Dopodiché c’è **Promozione**, identificata da un codice identificativo e un messaggio da mostrare agli utenti specificati all’interno della tabella **Promotion\_user**, che definisce appunto a quale utente deve venire mostrato quale promozione.

Il risultato di questa progettazione è il seguente database:

* Game (room, name)
* Gender (name)
* Media (url, game\_name, room\_location, type)
* Promotion (id, message)
* Promotion\_user (user\_email, promotion\_id)
* Room (location)
* Type (name)
* User (name, surname, street, house\_number, zip\_code, city, email, phone\_number, gender, password, type, verified, admin)

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

## Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gannt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Grazie a questo progetto abbiamo imparato almeno le basi dell’utilizzo di software che vengono usati anche nel mondo del lavoro per l’integrazione continua e i test, come Selenium e Jenkins.

# Bibliografia

## Sitografia

* <https://medium.com/@khandelwalnidhi/jenkins-setup-for-php-unit-testing-on-aws-c39baad7a99e>, *Jenkins Setup for PHPUnit*, 13.02.2019
* <https://stackoverflow.com/questions/39621263/jenkins-fails-when-running-service-start-jenkins>, *Jenkins fail to run,* 13.02.2019
* <https://www.ubuntu.com/download/server>, *Install ubuntu server,* 13.02.2019
* <https://thishosting.rocks/install-php-on-ubuntu/>, *Install PHP on ubuntu,* 13.02.2019
* <https://linuxize.com/post/how-to-install-jenkins-on-ubuntu-18-04/>,  *Install jenkins on ubuntu,* 13.02.2019
* <https://notifyjs.jpillora.com/>, Notify.js, 13.03.2019
* <https://tecadmin.net/setup-selenium-chromedriver-on-ubuntu/>, *Install and setup Selenium,* 15.02.2019
* <https://www.tutorialspoint.com/selenium/selenium_webdriver.htm>, *Selenium Webdriver,* 15.03.2019
* <https://www.seleniumhq.org/projects/webdriver/>, *Selenium Webdriver,* 15.03.2019
* <https://www.built.io/blog/run-selenium-tests-in-headless-browser>, *Run Selenium Tests In Headless Browser - Built.io Blog,* 15.03.2019
* <https://wiki.saucelabs.com/>*, Getting Started with Selenium for Automated Website Testing - The Sauce Labs Cookbook - Sauce Labs Documentation Wiki,* 15.03.2019
* <https://www.tutorialspoint.com/groovy/groovy_unit_testing.htm>, Groovy Unit Testing, 15.03.2019
* <https://maven.apache.org/install.html>, *Maven - Installing Apache Maven*, 20.03.2019
* <http://chromedriver.chromium.org/>, *ChromeDriver - WebDriver for Chrome*, 20.03.2019
* <https://github.com/PHPMailer/PHPMailer>, PHPMailer/PHPMailer: The classic email sending library for PHP, 22.03.2019
* <https://www.pexels.com/search/casino/>, 30+ Interesting Casino Photos · Pexels · Free Stock Photos, 22.03.2019
* <https://junit.org/junit5/>, JUnit 5, 29.03.2019

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o Qdc
* Prodotto
* …