Gestione Appartamenti

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

**Titolo:** Gestione Appartamenti

**Allievi**: Mattia Toscanelli, impiegato nello svolgimento del progetto.

**Classe**: I4AC

**Docenti**: Massimo Sartori

**Sezione scuola**: Scuola Arti e Mestieri Trevano, Informatica

**Data inizio**: 23.01.2020

**Fine**: 06.04.2020

## Abstract

E’ una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.*

## Scopo

Lo scopo di questo progetto è quello di realizzare un sito web che permetta di gestire l’affitto di appartamenti in determinati luoghi di villeggiatura. Gli utenti registrati a questo sito avranno la possibilità di affittare giornalmente o definitivamente un appartamento oppure ne potranno registrare di nuovi. Gli appartamenti potranno essere riservati sia da parte degli utenti registrati al sito oppure dal proprietario. Ogni riservazione dovrà essere confermata da un responsabile. Per ogni appartamento messo a disposizione dai proprietari dell’immobile si dovranno avere dei resoconti settimanali, mensili e annuali. In seguito sarà presente una pagina principale dove l’amministratore potrà inserire foto, inserire notizie e modificare il testo. Infine il sito web dovrà essere visualizzabile da tutti i principali browser e cellulari.

# Analisi

## Analisi del dominio

Bisogna trovare un metodo che permetta di gestire l’affitto di appartamenti in determinati luoghi di villeggiatura. L’obiettivo di questo progetto è dunque quello di creare un applicativo web che concede alle persone registrate al sito di poter cercare appartamenti da affittare oppure aggiungerne di nuovi. Esistono sul mercato già prodotti simili che svolgono queste funzioni ma che si differenziano per dei piccoli particolari. Infatti, grazie a questo progetto, i proprietari di appartamenti potranno visualizzare un riepilogo settimanale, mensile o annuale dei propri appartamenti.

Quando un utente di registra al sito dovrà verificare la propria e-mail per poter incominciare ad utilizzare l’applicativo. A questo punto l’utente avrà la possibilità di incominciare ad affittare appartamenti. Se invece l’utente vuole inserire dei propri annunci, bisognerà aspettare finché l’amministratore del sito abbia verificato quest’ultimo.

I compiti dell’amministratore non finiscono qui, infatti esso dovrà poter gestire una pagina principale dove vengono mostrare delle foto e delle notizie riguardanti il sito.

L’interfaccia dell’applicazione deve essere molto semplice e intuitiva, in poche parole deve essere comprensibile anche al meno esperto di informatica. Inoltre l’applicazione dovrà essere visualizzabile da tutti i principali browser e dispositivi mobili.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Sito web per la gestione degli appartamenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ci deve essere una pagina principale accessibile a tutti gli utenti, anche a coloro che non si sono registrati. |
| **002** | Ogni funzione del sito non deve presentare imprevisti. |
| **003** | Il sito deve essere responsive, cioè deve essere adattato a qualsiasi tipo di schermo. |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Pagina di registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Tutti possono accedere alla pagina di registrazione. |
| **002** | La e-mail dell’utente che si vuole registrare al sito deve essere verificata e non dove appartenere a un dominio di e-mail temporanee. |
| **003** | I dati obbligatori da inserire nella registrazione sono 6:  Email, password, nome, cognome, indirizzo e numero di telefono mobile. |
| **004** | C  Numero di telefono di ufficio e numero di telefono privato. |
| **ID: REQ-003** | |
| **Nome** | Pagina di login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Tutti gli utenti che vogliono accedere devono dapprima effettuare la verifica della e-mail. |
| **002** | Ci deve essere la possibilità di cambiare la password in caso di smarrimento. |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Pagina principale |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Si devono mostrare le notizie del sito. |
| **002** | Vengono mostrate del foto. |
| **003** | Solo gli amministratori della pagina possono aggiungere notizie, aggiungere foto e modificare il testo della pagina principale. |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Pagina di ricerca di appartamenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ci deve essere una pagina principale dove vengono mostrati tutti gli appartamenti. |
| **002** | La pagina deve avere una barra di ricerca per permettere di filtrare tutti gli appartamenti. |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Pagina singola di un determinato appartamento. |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ogni appartamento deve presentare le seguenti informazioni:  Citta, via, foto, pianta dell’appartamento, numero di locali, se offre posteggi o no, prezzo, spese, accessori, ammobiliato, se si può fumare o no, se si possono tenere animali, se si possono tenere bambini nell’appartamento o no, l’ubicazione e la carta, se è disponibile o meno e infine i commenti. |
| **002** | I proprietari dell’appartamento possono visualizzare un resoconto settimanale, mensile o annuale. |
| **003** | Ogni appartamento può essere riservato sia dal proprietario sia da un utente registrato. |
| **004** | Per ogni riservazione va scelto un periodo e quest’ultima deve essere accettata dal proprietario. |
| **005** | I proprietari devono avere la possibilità di aggiungere una riservazione per un cliente che magari lo chiama per telefono. |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Pagina di gestione dei propri appartamenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ci deve essere una pagina principale dove vengono mostrati tutti gli appartamenti. |
| **002** | La pagina deve avere una barra di ricerca per permettere di filtrare i propri appartamenti. |
| **003** | Si deve poter inserire nuovi appartamenti |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | Pagina di amministrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Questa pagina è accessibile solamente dall’amministratore. |
| **002** | C’è una tabella dove vengono mostrati tutti gli utenti registrati. |
| **003** | Ci deve essere la possibilità di modificare o eliminare un utente. |
| **004** | Ci deve essere una sezione per la modifica della pagina principale. |

## Use case

Questo è lo Use case (casi di utilizzo) per l’applicazione Gestione Appartamenti:

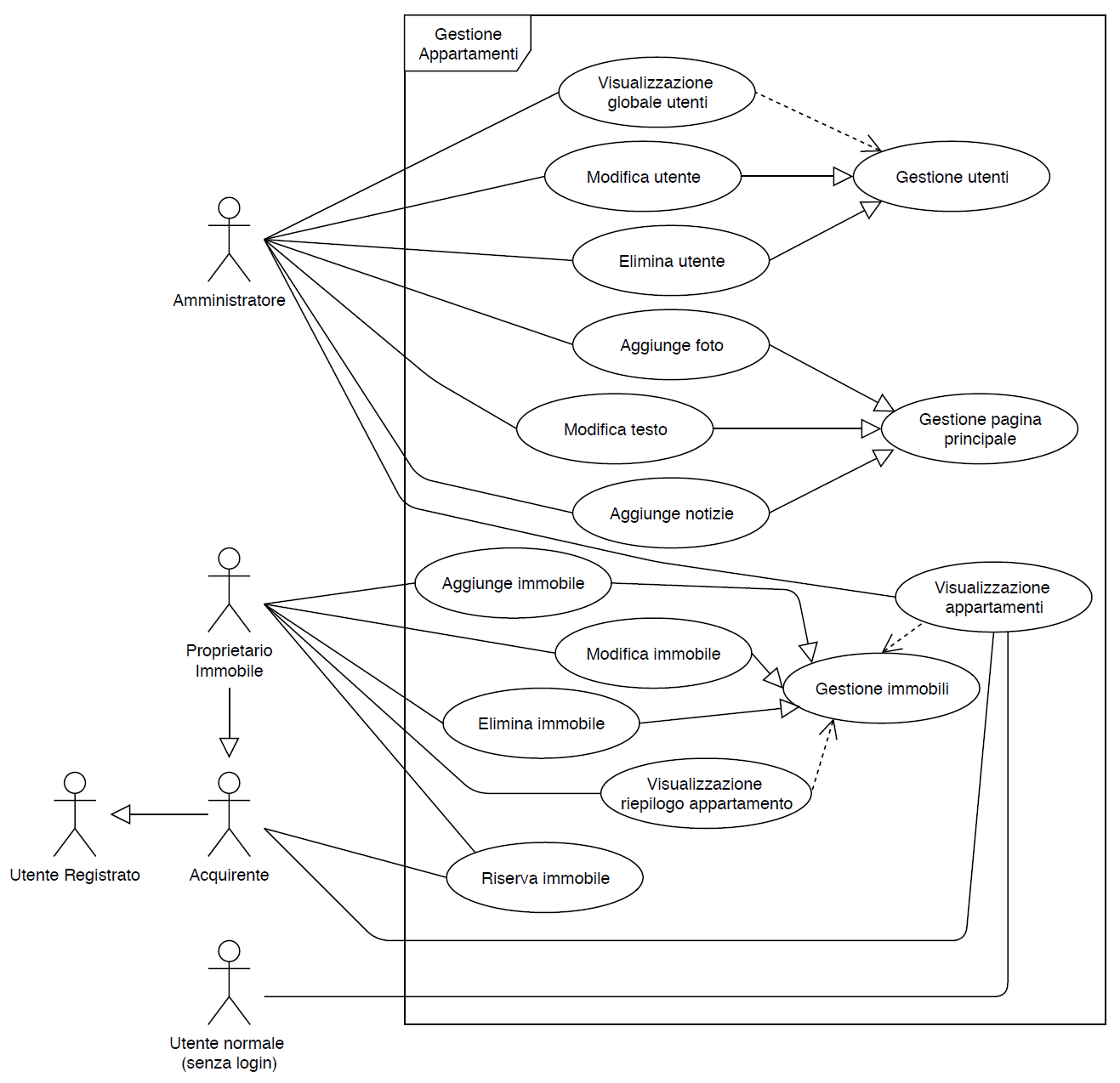


Figura 1 Use Case

Lo schema inizia con quattro tipologie di scenari: Amministratore (cioè colui che gestisce tutta l’applicazione), Proprietario immobile, Acquirente e Utente normale.

L’amministratore ha la possibilità di visualizzare una situazione globale di tutti gli utenti registrati al sito, mostrando tutti i dati non sensibili. Grazie a questa visualizzazione esso può modificare o eliminare un utente.

Inoltre l’amministratore può aggiungere foto, aggiungere notizie e modificare il testo nella pagina principale dell’applicazione.

Quando un utente si registra al sito, bisognerà verificare la propria e-mail. Una volta verificata questo utente diventa automaticamente un acquirente dell’applicazione. Esso può visualizzare tutti gli appartamenti messi a disposizione da altri e effettuare delle riservazioni.

L’utente proprietario immobile assume tutte le funzioni di un acquirente ma ha la possibilità di aggiungere, modificare o eliminare un immobile. Inoltre potrà visualizzare un riepilogo settimanale, mensile o annuale dei propri appartamenti. Per essere un proprietario immobile viene richiesta una verifica da parte dell’amministratore.

Infine esiste l’utente normale che non ha bisogno di una registrazione. Esso ha solamente la possibilità di visualizzare tutti gli appartamenti del sito.

## Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

|  |
| --- |
| gantt  Figura 2: Esempio di diagramma di Gantt. |

Se si usano altri metodi di pianificazione (es scrum), dovranno apparire in questo capitolo.

## 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

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che hw sarà disponibile durante lo sviluppo?

# 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

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

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo,

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# 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
* …