Cagnavin

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 6

2.4 Pianificazione 6

2.5 Analisi dei mezzi 6

2.5.1 Software 6

2.5.2 Hardware 6

3 Progettazione 7

3.1 Design dell’architettura del sistema 7

3.2 Design dei dati e database 7

3.3 Design delle interfacce 7

3.4 Design procedurale 7

4 Implementazione 8

5 Test 8

5.1 Protocollo di test 8

5.2 Risultati test 9

5.3 Mancanze/limitazioni conosciute 9

6 Consuntivo 9

7 Conclusioni 9

7.1 Sviluppi futuri 9

7.2 Considerazioni personali 9

8 Bibliografia 9

8.1 Bibliografia per articoli di riviste: 9

8.2 Bibliografia per libri 9

8.3 Sitografia 10

9 Allegati 10

# Introduzione

## Informazioni sul progetto

Questo progetto ha inizio in data 03.01.2025 e terminerà in data 04.04.2025.

Ho scelto di utilizzare la metodologia Waterfall ed utilizzerò Vaadin, un framework open-source per sviluppare applicazioni web moderne e interattive in Java, con una forte integrazione tra frontend e backend.

## Abstract

*Per questo progetto, ho deciso di creare un gestionale per il gruppo gastronomico Cagnavin. Prima di realizzare questo progetto, la gestione del gruppo e delle sue gite avveniva esclusivamente tramite Facebook, un metodo che risultava limitante e poco pratico. L'obiettivo del progetto è risolvere questa problematica, offrendo un sistema che consenta a chiunque di farsi un'idea più chiara delle attività del gruppo, senza la necessità di avere un account. In questo modo, si favorisce l'aumento della visibilità del gruppo e la possibilità di attrarre nuovi partecipanti.*

## Scopo

Lo scopo del progetto è sviluppare una piattaforma web moderna e intuitiva per il gruppo Cagnavin, che semplifichi e migliori la gestione delle gite gastronomiche organizzate, offrendo una soluzione più efficiente rispetto all'uso di Facebook. Il gestionale consentirà agli amministratori di organizzare, monitorare e gestire ogni aspetto delle gite, inclusa la creazione di eventi, la gestione delle iscrizioni, il caricamento di foto e la gestione dei pagamenti tramite email. Gli utenti, d’altra parte, avranno la possibilità di interagire con il sito in modo semplice, accedere facilmente alle informazioni relative alle gite passate e future, personalizzare il loro profilo e ricevere notifiche importanti.

L'obiettivo è rendere il processo di iscrizione alle gite più accessibile e comodo, favorendo la partecipazione di nuovi membri e migliorando la comunicazione tra amministratori e partecipanti. Inoltre, la piattaforma avrà un impatto positivo sulla gestione delle informazioni, consentendo a tutti i membri del gruppo di accedere facilmente alle foto e dettagli delle gite passate, oltre a migliorare l’esperienza complessiva, favorendo una maggiore organizzazione e soddisfazione degli utenti.

# Analisi

## Analisi del dominio

L’applicativo verrà usato da due categoria di utenti, gli amministratori, che si occupano di tutto quello che riguarda la gestione della gita e gli utenti, che possono essere sia dei partecipanti o semplici visitatori della pagina.

Principalmente si vuole facilitare l’iscrizione e la gestione delle gite del gruppo, che al momento come unica alternativa presenta il gruppo facebook che non permette una gestione ottimale ed automatica.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Login e registrazione account utente |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Possibilità di Login e registrazione all’interno del sito |
| **Sotto requisiti** | |
| **001** | Login con Nome Utente o Email e Password |
| **002** | Registrazione con Nome utente, Email, Password, Conferma Password |
| **003** | Password almeno di 8 caratteri con almeno una Maiuscola e almeno un carattere speciale |
| **004** | Codifica della password nel DB |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Sito Web – Pagina home |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Informazioni riguardanti la pagina iniziale del progetto |
| **Sotto requisiti** | |
| **001** | Il sito web deve essere responsive (computer, telefono, tablet) |
| **002** | Il sito web dovrà avere una pagina di login ed una di registrazione |
| **003** | Sarà una classica pagina di benvenuto, con alcune foto ed informazioni sul gruppo |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Sito Web – Storico delle gite |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Informazioni riguardante la pagina dello storico di tutte le gite |
| **Sotto requisiti** | |
| **001** | La pagina non necessita di essere autenticati per essere accessibile |
| **002** | Mostra tutte le gite passate, con le annesse informazioni e fotografie caricate |
| **003** | Lo storico si aggiorna in automatico, quando la data di una gita è passata, si potrà visualizzare in automatico nello storico |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | Modalità di pagamento |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Informazioni per la gestione del pagamento |
| **Sotto requisiti** | |
| **001** | Al momento dell’iscrizione, l’utente può selezionare se paga in contante o via TWINT |
| **002** | Selezionando il pagamento in contanti, viene inviata in automatico una fattura con un resoconto |
| **003** | Selezionando il pagamento via TWINT, viene inviata una mail con annesse coordinate di pagamento |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | Annullamento e rimborso |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Informazioni per la gestione dei rimborsi in caso di annullamento |
| **Sotto requisiti** | |
| **001** | Non è previsto nessun rimborso in caso il partecipante disdica |
| **002** | In caso la gita venga annullata il rimborso per chi ha pagato via TWINT avverrà in automatico |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | Regole per poter partecipare |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Informazioni per quanto riguarda i requisiti per poter partecipare alla gita |
| **Sotto requisiti** | |
| **001** | L’utente deve essere registrato |
| **002** | In caso non sia registrato, nel tentativo di richiesta verrà reindirizzato alla pagina |
| **003** | L’amministratore avrà una pagina dedicate alla gestione dei partecipanti |

**Spiegazione elementi tabella dei requisiti:**

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

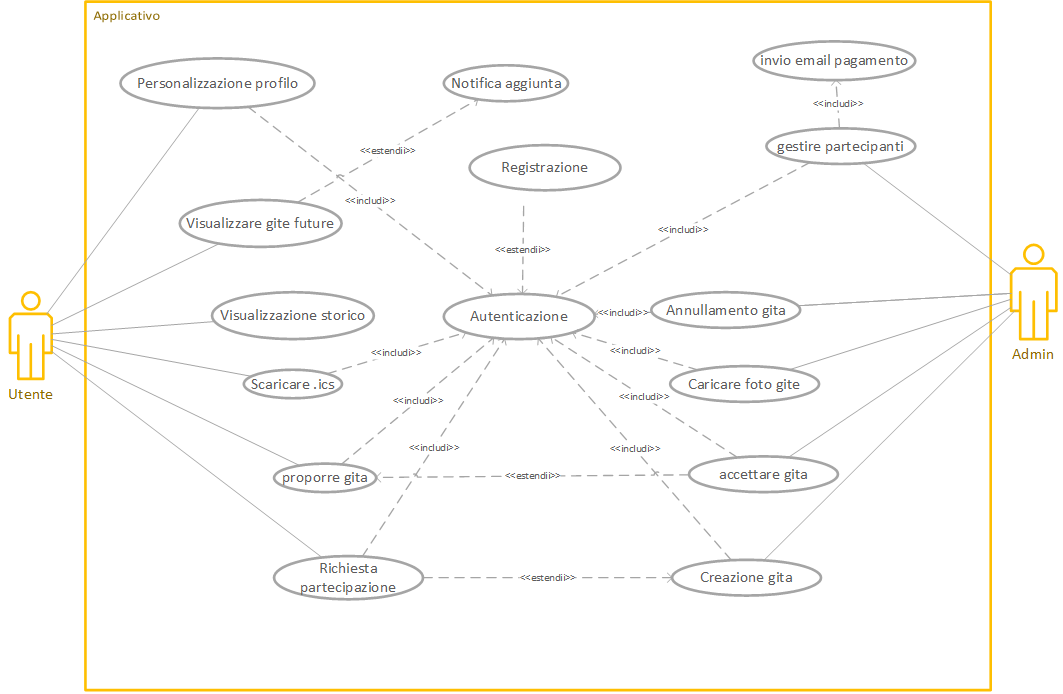
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case



All’interno dello Use Case esistono due attori, l’amministratore e l’utente normale. La pagina di login permette anche di registrarsi in caso non si abbia ancora un account.

Senza essersi registrato l’utente potrà visualizzare la pagina home contenente tutte le informazioni sul gruppo e avrà accesso allo storico delle gite e le gite future.

Una volta autenticato avrà la possibilità di partecipare alla gita, scaricare il calendario delle gite future, iscriversi alla newsletter ed avrà a sua disposizione una pagina per quello che riguarda la gestione del suo account. Inoltre potrà inviare una proposta di meta per una futura gita.

D’altro canto l’admin invece avrà a sua disposizione un pannello per la creazione di nuove gite, potrà accettare le gite proposte dagli utenti, che se accettate verranno aggiunte in automatico e

## 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:

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

Se si usano altri metodi di pianificazione (p.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 Gantt 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
* …