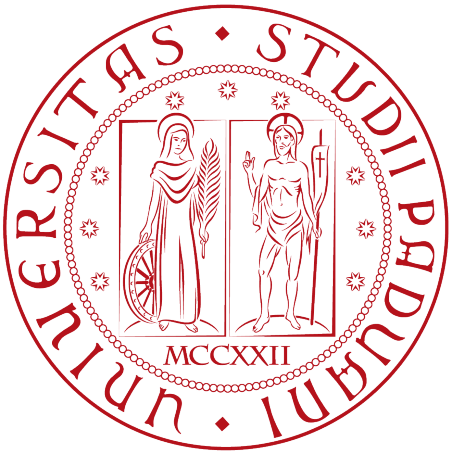
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea in Informatica



CS-Template: una applicazione per la piattaforma Zendesk basata su moderne tecnologie web

Tesi di laurea triennale

Relatore

Prof.Francesco Ranzato

Laureando

Singh Parwinder

Anno Accademico 2017-2018

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage(dal 03/06/2018 al 03/08/2018), della durata di circa 320 ore, dal laureando Singh Parwinder presso l'azienda Nextep alla sede di Cittadella.
In questo documento verrano descritte in dettaglio l'analisi dei requisiti, la progettazone, l'implementazione e la validazione dell'applicazione CS-template. L'applicazione in questione è stato realizzato utilizzanod le tecnolgie web inovvative sia per quanto riguarda lato frant-end dell'applicazione che quello back-end.
L'intero lavoro è stato svolto in ambiente Linux Ubuntu 18.04 LTS. Tutti i diagrammi delle classi, dei package e dei casi d'uso (presenti nei Capitoli 3 e 4) sono conformi allo standard UML 2.0. Per realizzarli è stato usato il software Astah Professional.

Il primo capitolo descrive l'azienda e il progetto di stage assegnato.

Il secondo capitolo descrive e tecnologie utlizzate durante tutto il periodo di stage.

Il terzo capitolo formalizza tutti i casi d'uso e i requisiti ad alto livello raccolti in fase di analisi dei requisiti.

Il quarto capitolo illustra ad alto livello la progettazione.

Il quinto capitolo approfondisce ...

Il sesto capitolo approfondisce ...

Nel settimo capitolo descrive ...

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

* gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
* per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*[g];
* i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

"Life is really simple, but we insist on making it complicated"

— Confucius

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. NomeDelProfessore, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare con affetto i miei genitori per il sostegno, il grande aiuto e per essermi stati vicini in ogni momento durante gli anni di studio.

Ho desiderio di ringraziare poi i miei amici per tutti i bellissimi anni passati insieme e le mille avventure vissute.

Padova, settembre 2018 Singh Parwinder

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

In questo capitolo viene brevemente descritta l'azienda ospitante in cui è stata svolta l'attività di stage e una descrizione ad alto livello del progetto realizzato.

1.1 Dominio aziendale

1.1.1 L'azienda ospitante

Nextep è una società fondata nel 2000 da Marco De Toni e Mirco Soffia, con sede attuale a Cittadella (PD). Opera nel settore informatico e si occupa di servizi web, web marketing e di infrastrutture per gestire le informazioni delle aziende, e più in generale ha come obiettivo quello di migliorare l'efficacia delle strategie di comunicazione web, delle aziende, dedicando particolare attenzione alla reputazione e all'identità digitale.

Nextep fa parte del gruppo Allos, insieme ad Allos Italia, Allos Sud Africa, Allos USA e Zero12. Allos si occupa di progetti e tecnologie per lo sviluppo del capitale umano, mentre Zero12 si occupa dello sviluppo di soluzioni mobile e cloud based. Il gruppo Allos è stato recentemente acquisito da EOH Holdings Ltd, una grande società sudafricana.

Nextep ha un organico di circa venti persone, tra dipendenti e collaboratori, con varie competenze: grafici, sviluppatori, esperti di web marketing e tecnici. Sono presenti tre gruppi principali di lavoro: quello di sviluppo, quello creativo e quello del supporto tecnico. In Nextep c'è un ambiente di lavoro giovane, dinamico ma allo stesso tempo professionale, ed è incentivata la collaborazione e la condivisione di conoscenze e idee tra le persone. Tutto questo favorisce sia la crescita individuale, dal punto di vista professionale, che la crescita e l'amalgamazione dei vari gruppi di lavoro.

Figura 1.1: Logo di Nextep: immagine tratta dal sito dell'azienda

1.1.2 Prodotti e servizi

Nextep lavora per clienti di diversa tipologia e conformazione, dalla piccola impresa privata alla multinazionale che si sta espandendo ulteriormente, e con questo offre svariati prodotti e servizi in base alle esigenze e alle opportunità del mercato e proprie.

La maggior parte dei progetti riguarda la realizzazione di portali e siti web, ma vengono sviluppati anche diversi altri prodotti, tra cui soluzioni e-commerce e applicazioni mobile, sviluppo di progetti di virtualizzazione, e storage networking. Inoltre negli ultimi mesi l'azienda si sta dedicato molto anche ai prodotti di machine learning, come i chatbot.

Nextep offre diversi tipi di servizi tra questi l'installazione e assistenza del portale di customer service Zendesk. Guida le diverse società(piccole o grandi) verso la gestione dei proprio cliente in manienra semplice ed efficace.

1.2 Lo stage

Il progetto di stage è consistito principalmente nella realizzaizone di una applicazione per la piattaforma di customer service Zendesk. La piattaforma Zendesk permette ad un'aziedna di gestire tutte le richieste(in sottoforma di tickets) di propri clienti in unico posto.
L'applicazione realizzata permette agli agenti(persone che gestiscono le richieste dei clienti) e agli aministratori di Zendesk di realizzare dei contenuti(chiamati template) HTML e CSS in maniera molto semplice e veloce, ovvero utilizzando un editor drag-and-drop. I template successivamente sono utilizzati nelle risposte verso i clienti. Questo permette di risparmiare una notevole quantità di tempo e non è neccario avere conoscenza di HTML e CSS. Diverse aziende(clienti di Nextep) avevano fatta la rischiesta esplicitamente di tale applicazione. Semplicemnte anche per rispondere agli utenti con le risposte molto "decorate" e velocizzare la generazione di codice html.

Capitolo 2

Tecnologie utilizzate

In questo capitolo seguirà un elenco delle tecnologie di riferimento adottate per la realizzazione dell'applicazione CS-Template.

2.1 Frontend

2.1.1 Angular 6

Angular è una piattaforam opensourse realizzato da Google nel 2016 che permette di creare le SPA, sfruttando i pattern architetturali MVC e MVVM.

Le applicazioni sviluppate in Angular vengono eseguite interamente dal web browser dopo essere state scaricate dal web server. Questo comporta il risparmio di dover spedire indietro la pagina web al web-server ogni volta che c'è una richiesta di azione da parte dell'utente. Il codice generato da Angular gira su tutti i principali web browser moderni.

Ogni pagina web viene costruitta da diversi componenti. Un componente in Angular in generale è una piccola parte della view che reppressenta una specifica funzionalità(esempio la navbar). Ogni component ha una propria logica strutturale (scritta tramite appositi marcatori HTML), di presentazione (scritta con appositi fogli di stile CSS oppure SCSS) e di business (scritta con il linguaggio di programmazione TypeScript). Tutti i componenti possono comunicare tra di loro scambiandosi oggetti, lo scambio viene fatto utilizzando diversi strumenti messi a disposizione da Angular. Oggi tale framework è alla versione 6.

Figura 2.1: Logo di Angular

2.1.2 Angular material

Angular material è una libreria che contiene una raccolta di componenti di Materia DesignG. Questo libreria è stata sviluppata sempre da Google e permette di realizzare delle UI molto avvanzata in maniena molto semplice. Fornendo una serie di semplici componenti(come i pulsanti, inputbox ecc) di Angular già fatta permette algli sviluppatori di risparmiare una notevole quantità di tempo. Tutti i componenti sono testati da Google garantendo cosi un corretto funzionamento.

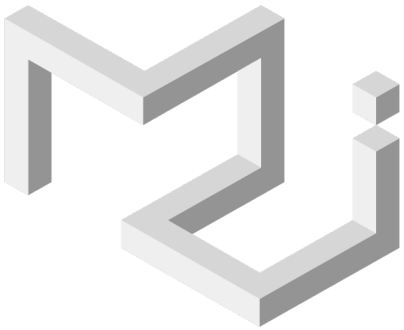


Figura 2.2: Logo di Material UI

2.1.3 Typescript

TypeScript è un linguaggio di programmazione libero ed Open source sviluppato da Microsoft basato su EMCAScript 6. Esso estende la sintassi di Javscript aggiungendo il concetto di tipizzazione(interfacce, classi, enum ecc). Questo lo rende molto simile ai linguaggi di programmmazione come Java oppure C++, e diventa anche molto semplice l'applicazione di molti design pattern conosciuti.

TypeScript nasce dal crescente bisogno di un linguaggio front-end per lo sviluppo di applicazioni JavaScript larga scala. Il linguaggio è nato dalla necessità di sicurezza e robustezza, sia da sviluppatori interni a Microsoft sia clienti.

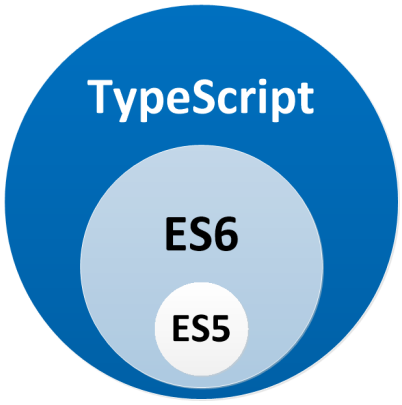


Figura 2.3: Typescript rispetto ES6 e ES5

2.1.4 HTML 5

L'HTML5 è un linguaggio di markup per la strutturazione delle pagine web, pubblicato come W3C Recommendation da ottobre 2014. HTML è stato usato come linguaggio per la definizione della logica strutturale del front-end dell'applicazione. Una delle principali vantaggi di Angular sta proprio nel utilizzo di HTML pure rispetto i framework/librerie revali come React oppure Vue.

Figura 2.4: Logo di HTML 5

2.1.5 SASS

Sass (Syntactically Awesome StyleSheets) è un'estensione del linguaggio CSS che permette di utilizzare variabili, di creare funzioni e di organizzare il foglio di stile in più file.

Il linguaggio Sass si basa sul concetto di preprocessore CSS, il quale serve a definire fogli di stile con una forma più semplice, completa e potente rispetto ai CSS e a generare file CSS ottimizzati, aggregando le strutture definite anche in modo complesso. SASS/CSS è un linguaggio utilizzato per definire la presentazione dell'applicazione. Poichè Angular Material fornisce molti componenti prefatti, questo linguaggio è utilizzato principalmente per definire il layout delle pagine.

Figura 2.5: Logo di SASS

2.2 Tecnologie Backend

Per quanto riguarda il lato Backend è stato deciso di utilizzare il cloud computing. Esso è la distribuzione di servizi di calcolo, come server, risorse di archiviazione, database, rete, software, analisi e molto altro, tramite Internet ("il cloud").

Le società che offrono questi servizi di calcolo sono dette provider di servizi cloud e in genere addebitano un costo per i servizi di cloud computing in base all'utilizzo(on demand). I provider più famosi oggigiorno sono Amazon, Microsoft e Google.

Dopo una breve analisi e confronto tra i servizi offerti da questi provider è stato deciso di utilizzare i servizi offerti da Amazon, ovvero gli AWS. La scelta è stata fatta soprattutto per la pololarità dei servizi Amazon, essendo Amazon primo a fornire questo tipo di servizio sono attualmente molto più utilizzati rispetto gli altri due. Per quanto riguarda i prezzi, quantità e qualità di servizi offerti da tuti i provider non è stato trovato un rirevante differenza.

Figura 2.6: AWS cloud computing

2.2.1 Amazon Web Services

Per l'applicazione in questione sono stati identificati specifici servizi di Amazon, che hanno permesso poi di realizzare un'architettura serverless facilmente scalabile. In seguito è fornita una descrizione generale dei servizi scelti.

2.2.2 AWS API Gateway

Amazon API Gateway è un servizio il cui scopo è quello di semplificare agli sviluppatori la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione delle API su qualsiasi scala. Questo servizio permette di creare un punto d'ingresso attraverso il quale le applicazioni posso accedere ai dati. Fornisce una API per ricevere le richiesta dalle applicazioni, dopo ogni richiesta questo servizio genera un evento che esegue qualcosa(una funzione lambda, un altra chiamata ecc).

Figura 2.7: Logo API Gateway

2.2.3 AWS Lambda

Sono delle semplici funzioni che ricevano come input un evento e possono ritornare qualche valore. Possono essere scritte in molti linguaggii dprogrammazione, come Nodejs(javascript), Java, C++ e Phython.

Queste funzioni possono comunicare con qualsiasi servizio di Amazon utilizzando AWS-SDK. In questo progetto sono state utilizzate diverse funzioni di questo tipo, princiaplamente per scrivere o leggere i dati su database.

Figura 2.8: Logo di AWS Lambda

2.2.4 AWS DynomoDB

Amazon DynamoDB è un database non relazionale che fornisce prestazioni affidabili su qualsiasi scala. Si tratta di un database multi-master, multiregione e completamente gestito che fornisce latenza costante di pochi millisecondi e sicurezza integrata, backup e ripristino e cache in memoria.

Tutti i dati degli utenti e i tempalte sono stati salvati su questo database. L'acceso a tali avviene solo tramite le funzioni lambda.

Figura 2.9: Logo di AWS DynamoDB

2.2.5 AWS S3

Amazon Simple Storage Service è uno storage di oggetti creato per memorizzare e ripristinare qualsiasi volume di dati da qualunque origine: siti web, applicazioni per mobile o dati provenienti da diversi dispositivi. Può essere utilizzato per memorizzare file multimediali ed è ideale per acquisire dati quali foto, video o immagini di risoluzione elevata da dispositivi mobili, backup di dispositivi mobile o computer.

Nel contesto dello stage è stato utilizzato per garantire la persistenza di tutti i dati non strutturati dell'applicazione. Inoltre l'applicazione stassa è stata caricata su questo servizio.

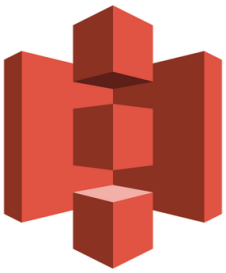


Figura 2.10: Logo di AWS S3

2.2.6 AWS Cognito

Amazon Cognito permette di aggiungere strumenti di registrazione, accesso e controllo degli accessi alle app Web e per dispositivi mobili in modo rapido e semplice. Amazon Cognito permette di ricalibrare le risorse per milioni di utenti e supporta l'accesso con provider di identità social quali Facebook, Google e Amazon.

Nel contesto dello stage è stato utilizzato per garatire l'accesso alla pagina di amministratore solo agli utenti con le credenziali valide.

Figura 2.11: Logo di AWS Cognito

2.3 Tecnologie di supporto

2.3.1 Git

Git è un software di controllo di versione distribuito, creato nel 2005 da Linus Torvalds (creatore di Linux). È usabile principalmente da interfaccia linea di comando, ed oggi è in assoluto uno dei software per il controllo di versione più utilizzati. Come tutti i software per il controllo di versione, si basa sul concetto di repository, ovvero un ambiente in cui vengono immagazzinati i metadati che possono essere recuperati e aggiornati da chi può averne accesso (è possibile ripristinare versioni precedenti dei dati caricati nel repository, poichè essi non vengono sovrascritti con gli aggiornamenti).

Figura 2.12: Logo di Git

2.3.2 GitLab

GitLab è una piattaforma web open source che permette la gestione di repository Git. GitLab permette la creazione di repository pubblici o privati, in cui gli sviluppatori possono caricare il proprio codice e gestire le modifiche alle varie versioni in contemporanea al lavoro di più persone.

In GitLab è possibile lavorare parallelamente ad altre persone sullo stesso progetto senza generare conflitti, caricare il proprio lavoro nel repository remoto (operazione di push) e poter unire alla fine le modifiche di tutti in un unico progetto (operazione di merge). GitLab mette a disposizione diverse funzionalità a seconda del tipo di abbonamento e del prezzo pagato. È comunque possibile utilizzarlo gratuitamente, seppur con delle limitazioni. Nel contesto dello stage è stato utilizzato GitLab con l'account aziendale di Nextep, permettendo cosi di lavorare su una repository privata.

Figura 2.13: Logo di Gitlab

2.3.3 Atom

Atom è un editor di testo Open-Sourse sviluppato nel 2014 da GitHub. Questo editor è stato scritto completamente utilizzando le tecnologie web(Html, CSS e Javascript). Una delle cose molto interessanti di questo editor è quello di avere Git già integrato, permettendo in questo modo di fare commit sin da subito.

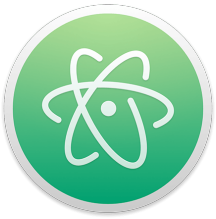


Figura 2.14: Logo di Atom

2.4 Zendesk Apps framework(ZAF)

ZAF è un semplice framework sviluppato da Zendesk Inc. Permette di integrare nella piattaforma Zendesk applicazione web realizzata con qualsiasi tecnologia web.

Questo framework genera dei package che rappressentono un app di Zendesk. Take package può essere installato sulla propria piattaforma Zendesk come una applicazioen privata oppure caricata(gratuitamente oppure a pagamento) su Market place di Zendesk rendendola cosi disponibile a tutti gli utenti nel mondo. Il package è formato da due file JSON(manifest.json) e una cartella assets.La cartella contiene le icone da visualizzare dopo l'installazione dell'applllicazione sulla piattaforma, mentre il file manifest.json contiene tutte le informazioni riguardo l'applicazione. Il seguente codice mostra la struttuta generale di un semplice file manifest.json.

```
{
  "name": "CS-Template",
  "author": {
        "name": "Singh Parwinder",
        "email": "mio@email.com",
  },
  "location": {
        "ticket_sidebar":  "https://www.paginaweb.org"
  },
  "parameters": [
  {
    "name": "token",
        "type": "text",
        "required": true
  }]
}
```

L'atrtibuto "location" è quello più importatnte. Questo attributo specifica dove visualizzare l'appllicazione sulla piattaforma dopo l'installazione. Si ha la possibilità di scegliere tra piu di dieci possizioni differenti, e per ogni posizione speficato bisgona indicare l'indirizzo web della pagina da visualizzare. Quindi in parole povere queste posizioni sono dei semplici iframe che mostrano le pagine web presenti su un server remoto. Per questo motivo le applicazioni possono essere realizzate con qualsiasi tecnologia web.

L'attributo "parameters" indica i parametri da chiedere all'utente durante l'installazione dell'applicazione sulla piattaforma. Nel contesto del applicazione è richiesto di inserire obbligatoriamente un token da 16 cifre, l'inserimento sbaglaito di tale valore comporta inutilizzo dell'applicazione CS-Template.

2.4.1 ZAF Client SDK

Ovviamente non avrebbe senso semplcemente visualizzare le pagine web sulla piattaforma se queste non possono interagire con le funzionalità di Zendesk.

Per permettere all'applicazione web di interagire con la piattaforma Zenedesk, è reso disponibile una libreria(ZAF Client SDK) da utilizzare nella propria applicazione web. Questa libreria fornisce un oggetto speciale(ZAFClient) con una ciquantina di metodi che permettono(una volta caricata la pagina nella piattaforma) di interagire con le diverse funzionalità di Zendesk.

Seguente codice invoca il metodo "invoke" dell'istanza "client" del oggetto ZAFClient per aggiungere testo "Hello World!" nella risposta da inviare all'utente.

```
let client = ZAFClient.init();
client.invoke('ticket.comment.appendText', 'Hello world!').then
    (function() {
       console.log('text has been appended');
});
```

Capitolo 3

Descrizione progetto ed analisi dei requisiti

Breve introduzione al capitolo

3.1 Introduzione al progetto

3.2 Analisi preventiva dei rischi

Durante la fase di analisi iniziale sono stati individuati alcuni possibili rischi a cui si potrà andare incontro. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

1. Performance del simulatore hardware
Descrizione: le performance del simulatore hardware e la comunicazione con questo potrebbero risultare lenti o non abbastanza buoni da causare il fallimento dei test.
Soluzione: coinvolgimento del responsabile a capo del progetto relativo il simulatore hardware.

3.3 Requisiti e obiettivi

3.4 Pianificazione

Capitolo 4

Analisi dei requisiti

Breve introduzione al capitolo

4.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo Unified Modeling Language (UML) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

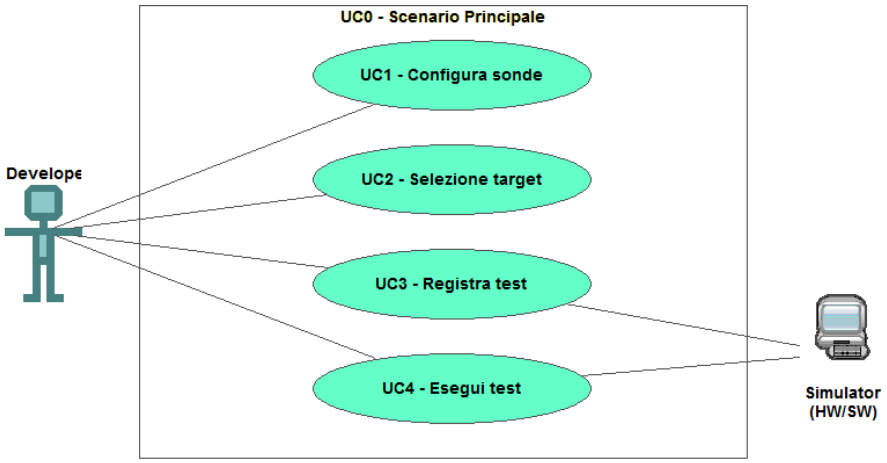


Figura 4.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

4.2 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e degli use case effettuata sul progetto è stata stilata la tabella che traccia i requisiti in rapporto agli use case.
Sono stati individuati diversi tipi di requisiti e si è quindi fatto utilizzo di un codice identificativo per distinguerli.
Il codice dei requisiti è così strutturato R(F/Q/V)(N/D/O) dove:

R = requisito

F = funzionale

Q = qualitativo

V = di vincolo

N = obbligatorio (necessario)

D = desiderabile

Z = opzionale

Nelle tabelle 4.1, 4.2 e 4.3 sono riassunti i requisiti e il loro tracciamento con gli use case delineati in fase di analisi.

Tabella 4.1: Tabella del tracciamento dei requisti funzionali

Requisito	Descrizione	Use Case
RFN-1	L'interfaccia permette di configurare il tipo di sonde del test	UC1

Tabella 4.2: Tabella del tracciamento dei requisiti qualitativi

Requisito	Descrizione	Use Case
RQD-1	Le prestazioni del simulatore hardware deve garantire la giusta esecuzione dei test e non la generazione di falsi negativi	-

Tabella 4.3: Tabella del tracciamento dei requisiti di vincolo

Requisito	Descrizione	Use Case
RVO-1	La libreria per l'esecuzione dei test automatici deve essere riutilizzabile	-

Capitolo 5

Progettazione e codifica

Breve introduzione al capitolo

5.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

5.2 Ciclo di vita del software

5.3 Progettazione

Namespace 1

Descrizione namespace 1.

Classe 1: Descrizione classe 1

Classe 2: Descrizione classe 2

5.4 Design Pattern utilizzati

5.5 Codifica

Capitolo 6

Verifica e validazione

Capitolo 7

Conclusioni

7.1 Consuntivo finale

7.2 Raggiungimento degli obiettivi

7.3 Conoscenze acquisite

7.4 Valutazione personale

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia