Travel

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 19/03/2023 | <1.0> | Am decis ce functionalitati va avea aplicatia. | Bode Andreea-Nicoleta |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Cuprins

[1. Introduction 4](#_Toc132783189)

[2. Non-functional Requirements 4](#_Toc132783190)

[2.1 Availability 4](#_Toc132783191)

[2.2 Performance 4](#_Toc132783192)

[2.3 Security 4](#_Toc132783193)

[2.4 Testability 5](#_Toc132783194)

[2.5 Usability 5](#_Toc132783195)

[3. Design Constraints 5](#_Toc132783196)

# Introduction

Documentul Supplementary Specification este o componentă importantă a cerințelor generale de sistem pentru un proiect software. Acesta oferă o privire de ansamblu asupra cerințelor de sistem care nu sunt surprinse în mod adecvat în cazurile de utilizare ale modelului de caz de utilizare. Aceste cerințe includ cerințe legale și de reglementare, cum ar fi standardele de aplicație la care software-ul trebuie să le respecte. În plus, documentul subliniază atributele de calitate pe care trebuie să le posede sistemul, inclusiv cerințele de utilizare, fiabilitate, performanță și suportabilitate. Alte cerințe, cum ar fi sistemele de operare și mediile, cerințele de compatibilitate și constrângerile de proiectare, sunt, de asemenea, incluse în specificația suplimentară. În mod colectiv, aceste cerințe oferă o imagine cuprinzătoare asupra cerințelor sistemului, asigurând că software-ul este dezvoltat pentru a satisface nevoile părților interesate și ale utilizatorilor săi.

# Non-functional Requirements

## Availability

Definiție: capacitatea sistemului de a fi operațional și accesibil atunci când este necesar.

Sursa stimulului: utilizatorul sau alt sistem care încearcă să acceseze sistemul.

Stimul: utilizatorul sau alt sistem încearcă să acceseze sistemul.

Mediu: starea curentă a sistemului când sosește stimulul.

Artefact: Componentele sistemului responsabile cu gestionarea cererilor utilizatorilor.

Răspuns: Sistemul oferă acces la resursele solicitate.

Măsura de răspuns: procentul de timp în care sistemul este disponibil într-o anumită perioadă, cum ar fi disponibilitatea de 99,9% pe an.

Tactici: redundanță, toleranță la erori, planificare de recuperare în caz de dezastru, întreținere proactivă și echilibrare a sarcinii.

## Performance

Definiție: capacitatea sistemului de a-și îndeplini funcțiile într-un interval de timp acceptabil și cu resurse adecvate.

Sursa stimulului: utilizatorul sau alt sistem care încearcă să acceseze sistemul.

Stimul: utilizatorul sau alt sistem efectuează o acțiune care necesită procesarea sistemului.

Mediu: starea curentă a sistemului când sosește stimulul.

Artefact: Componentele sistemului responsabile cu procesarea cererilor utilizatorilor.

Răspuns: Sistemul efectuează acțiunea solicitată într-un interval de timp acceptabil.

Măsura de răspuns: timpul de răspuns pentru finalizarea unei sarcini specifice sau numărul de solicitări procesate pe secundă.

Tactici: stocarea în cache, echilibrarea încărcăturii, optimizarea resurselor și procesarea paralelă.

## Security

Definiție: capacitatea sistemului de a preveni accesul neautorizat, de a proteja datele și de a menține confidențialitatea, integritatea și disponibilitatea sistemului și a datelor acestuia.

Sursa stimulului: un potențial atacator sau un utilizator care încearcă să acceseze sistemul.

Stimul: o încercare de încălcare a securității, cum ar fi o încercare de conectare neautorizată sau un atac de malware.

Mediu: starea curentă a sistemului când sosește stimulul.

Artefact: Componentele sistemului responsabile de securitate și controlul accesului.

Răspuns: Sistemul previne încălcarea securității sau se recuperează din aceasta și anunță părțile corespunzătoare.

Măsura de răspuns: numărul de încălcări de securitate reușite și nereușite și timpul necesar pentru detectarea și recuperarea după o încălcare.

Tactici: controlul accesului, criptarea, firewall-urile, detectarea și prevenirea intruziunilor și audituri de securitate.

## Testability

Definiție: ușurința cu care sistemul poate fi testat pentru a verifica funcționalitatea, performanța și calitatea acestuia.

Sursa stimulului: echipa de testare sau instrumente automate de testare.

Stimul: Un caz de testare sau o suită de teste este executată pe sistem.

Mediu: Mediul de testare și configurația.

Artefact: Componentele sistemului și cazurile de testare în curs de executare.

Răspuns: Sistemul returnează rezultatele așteptate și defectele sunt identificate și rezolvate.

Măsura de răspuns: procentul de cazuri de testare care trec sau eșuează, numărul de defecte găsite și rezolvate și timpul necesar pentru executarea cazurilor de testare.

Tactici: testare unitară, testare de integrare, testare automată și gestionare a cazurilor de testare

## Usability

Definiție: ușurința de utilizare și capacitatea de învățare a sistemului pentru utilizatorii săi vizați.

Sursa stimulului: Utilizatorul sau alte părți interesate implicate în sistem.

Stimul: utilizatorul interacționează cu sistemul sau oferă feedback cu privire la utilizarea acestuia.

Mediu: Mediul utilizatorului și contextul de utilizare.

Artefact: Componentele sistemului responsabile pentru interacțiunea și feedbackul utilizatorului.

Răspuns: Sistemul oferă o interfață de utilizator intuitivă și eficientă, iar feedback-ul este încorporat pentru a îmbunătăți gradul de utilizare.

Măsura de răspuns: rata de succes a sarcinilor utilizatorului, timpul de finalizare a sarcinilor și evaluările de satisfacție ale utilizatorilor.

Tactici: proiectare centrată pe utilizator, testare a gradului de utilizare, mecanisme de feedback și instruire și documentare pentru utilizatori.

# Design Constraints

Limbaje de programare: Sistemul va fi dezvoltat folosind Java pentru backend și React pentru frontend.

Framework: Spring va fi folosit ca cadru de backend.

Baza de date: PostgreSQL va fi folosit ca sistem de management al bazei de date.

Securitate: Sistemul va folosi protocoale sigure de autentificare și autorizare pentru a asigura securitatea informațiilor utilizatorului.

Scalabilitate: sistemul va fi proiectat pentru a gestiona un număr mare de utilizatori și date și va fi scalabil pentru a se adapta creșterii viitoare.

Performanță: Sistemul va fi optimizat pentru performanță ridicată, cu timpi de răspuns rapid și timpi de nefuncționare minimi.

Interfață cu utilizatorul: interfața cu utilizatorul va fi concepută pentru a fi ușor de utilizat și intuitivă, cu navigare clară și acces ușor la informații.

Integrare: sistemul va fi proiectat să se integreze cu servicii terțe, cum ar fi e-mailul și platformele de social media, pentru a-și îmbunătăți funcționalitatea și gradul de utilizare.

Cerințe de utilizare: sistemul ar trebui să fie proiectat având în vedere experiența utilizatorului și capacitatea de utilizare și poate fi necesar să respecte anumite linii directoare și standarde de proiectare.

Constrângeri bugetare: Designul sistemului trebuie să respecte orice constrângeri bugetare stabilite de organizație sau de părțile interesate de proiect, inclusiv limitări privind achizițiile de hardware și software.

Software process requirements: Procesul de dezvoltare pentru sistem este bazat pe standardele organizaționale sau cerințele proiectului.

Prescribed use of developmental tools: pentru Travel am ales ca sisteme de control al versiunilor Git-ul si ca mediu de dezolvotare integrat am ales Intellij pentru Back-end si Visual Code pentru Front-end.

Architectural and design constraints: sistemul să respecte constrângeri arhitecturale sau de proiectare, cum ar fi utilizarea unui sistem specific de gestionare a bazelor de date PostgreSQL sau includerea unor caracteristici de securitate specifice precum intrucerea parolei de doua ori pentru confirmare.