**Universitatea “Alexandru Ioan Cuza” Iași**

**Facultatea de Economie si Administrarea Afacerilor**

**Master Sisteme informaționale pentru afaceri**

**PROIECT**

**TESTAREA SI EVALUAREA APLICATIILOR INFORMATICE**

**Echipa 4:**

**PloaieArdela**

**Barjă Ana-Mădălina**

**Parlung Valy-Elena**

**Petrescu Petre**

**Perte Ardela-Ramona**

**Vrabie Oana**

**Stanca Clara-Ardela**

**Lambrino Mirela-Viorica**

**Cuprins**

[PLAN DE TESTARE 3](#_Toc35867619)

[II. Plan testare 31](#_Toc35867620)

[Istoric revizii 31](#_Toc35867621)

[2.1 Cod de identificare a planului de testare 31](#_Toc35867622)

[2.2 Referințe 31](#_Toc35867623)

[2.3 Introducere 31](#_Toc35867624)

[2.4 Elemente de testare 35](#_Toc35867625)

[2.5 Caracteristici ce trebuie testate 37](#_Toc35867626)

[2.6 Caracteristici ce nu trebuie testate 38](#_Toc35867627)

[2.7 Abordare 39](#_Toc35867628)

[2.8 Criterii de trecere/picare a testului de către elementele testate 41](#_Toc35867629)

[2.9 Criteriile de suspendare/reluare a procesului de testare 41](#_Toc35867630)

[2.10 Rezultate finale ale testelor 42](#_Toc35867631)

[2.11 Sarcini de testare 42](#_Toc35867632)

[2.12 Nevoi de mediu 42](#_Toc35867633)

[2.13 Responsabilităţi 43](#_Toc35867634)

[2.14 Nevoi de personal şi specializare 43](#_Toc35867635)

[2.15 Program 43](#_Toc35867636)

[2.16 Riscuri şi evenimente neprevăzute 44](#_Toc35867637)

[2.17 Aprobări 44](#_Toc35867638)

PLAN DE TESTARE

# II. Plan testare

## Istoric revizii

|  |  |  |  |
| --- | --- | --- | --- |
| **Nume** | **Data** | **Detalii (motivul schimbării)** | **Versiune** |
| PloaieArdela  Parlung Valy-Elena  Barjă Ana-Mădălina  Oana Vrabie  Ardela Perte  Petrescu Petre  Stanca Clara  Lambrino Viorica | 05.05.202x | Creare varianta initiala a documentului | 1.0 |
|  |  |  |  |
|  |  |  |  |

## Cod de identificare a planului de testare

**Cod STP**: *CT01.STP* – *STP pentru HotelFranco versiunea 1.0*

## Referințe

|  |  |
| --- | --- |
| **Aplicaţie informatică** | HotelFranco |
| **Domeniu** | Gestiunea generală a rezervărilor, utilizatorilor și a serviciilor interne desfășurate în cadrul hotelului Franco. |
| **Versiune** | 1.0 |
| **Mediul de dezvoltare** | Microsoft Visual Studio 2017 & Microsoft SQL Server. |

## Introducere

***HotelFranco*** este un produs software realizat cu scopul gestionării în mod personalizat a activităților desfășurate în cadrul hotelului Franco. Aplicația conține module destinate administrării utilizatorilor, gestionării rezervărilor, a sălilor de conferință, a serviciilor oferite de către hotel, a tarifelor și facilităților dar și o pagină destinată recenziilor oferite de clienți. Prin intermediul acestor module angajații au posibilitatea de a obține și prelucra toate informațiile necesare îndeplinirii tuturor atribuțiilor asignate. Totodată, aplicația dispune și de o serie de rapoarte grupate în funcție de modulele enumerate anterior: grad ocupare hotel, clienți fideli, camere disponibile, centralizator săli conferință, rezervări și servicii. Dobândirea abilităților necesare utilizării într-un mod optim a aplicației se realizează în cadrul training-ului oferit de către firma dezvoltatoare pe o perioadă de două săptămâni din momentul implementării soft-ului în centru.

**Scopul** acestui plan de testare constă în verificarea funcționării produsului software oferit și îndeplinirii de către acesta a tuturor specificațiilor regăsite în primul capitol al acestui document. În cazul identificării eventualelor neconcordanțe, acestea vor fi raportate echipei dezvoltatoare pentru a putea fi remediate. În cadrul structurii planului de testare vor fi evidențiate activitățile și elementele de testat, clasele de echivalență, bug-urile și calitatea produsului.

Aplicaţia poate fi utilizată de cinci categorii de useri: administrator, manager, client și vizitator. Administratorul are drepturi de utilizare depline a aplicației, iar celelate categorii au accesibilități conform celor descrise în capitolul 1.5.

Proiectul va conţine următoarele metode de testare:

* Testarea componentelor (instalării/dezinstalării)
* Testarea securităţii;
* Testarea pozitivă;
* Testarea negativă;
* Testarea interfeţei;
* Testarea valorilor limită;
* Testarea de traseu (flux);
* Testarea cerințe de business;
* Testarea documentației.
* Testarea White Box
* Testare Multibrowser
* Testare stress

**Testarea interfeţei**

Modalitatea de testare va ţine cont de specificul aplicaţiei. Se va testa interfaţa utilizator responsabilă de introducerea datelor, modul în care datele sunt transmise între modulele aplicaţiei.

Se va testa interfaţa utilizator responsabilă de parcurgerea prin meniuri, se va testa dacă interfaţa este sau nu prietenoasă, dacă butoanele şi mesajele de afişare sunt sugestive, cum se realizează navigarea prin meniuri şi posibilitatea modificarii dimensiunilor formularului.

**Testarea pozitivă**

Modalitatea de testare va ţine cont de specificul aplicaţiei. Testarea pozitivă are ca scop confirmarea funcţionalităţii generale. Se va pune accent pe cazurile de utilizare uzuale, ignorând cazurile excepţionale. Testarea pozitiva va verifica dacă sistemul face ceea ce trebuie sa facă. Testele se vor realiza prin introducerea de valori care se potrivesc cu cerinţele aplicaţiei şi se va urmări dacă aplicaţia returnează date valide. De asemenea se va verifica dacă funcţionalitatea de bază este conformă cu cerinţele utilizatorilor.

**Testarea negativă**

Modalitatea de testare va ţine cont de specificul aplicaţiei. Se vor căuta defectele aplicaţiei, punându-se accent pe cazurile de utilizare excepţionale. Testarea negativă va verifica dacă sistemul nu face ceea ce trebuie sa facă. Acesta este un tip de testare care este făcut pentru ca testerul să se asigure că sistemul funcţionează bine pentru intrări neprevăzute.

**Testarea valorilor limită**

Modalitatea de testare va ţine cont de specificul aplicaţiei. Se va testa programul cu valori de intrare de la limitele (spre interior şi spre exterior) claselor de echivalenţă.

**Testarea de traseu**

Modalitatea de testare va ţine cont de specificul aplicaţiei. În timpul execuţiei programul trece prin mai multe stări. Se vor urmări efectele fiecărei stări (condiţii creare, modificări în interfaţă). Selectarea tranziţiilor se face pe baza stării unei hărţi de tranziţii care indică toate stările posibile, declanşatorii fiecărei stări şi elementele fiecărei stări.

**Testarea cu script de testare**

Un script de test în testarea software-ului este un set de instrucţiuni care vor fi efectuate pe sistemul supus încercării pentru a testa functionalitatea aplicatiei. In acest caz trebuie descrisi foarte bine toti pasii ce sunt facuti pentru a obtine rezultatul dorit. De obcei test script este facut de un tester mai avansat.

**Testarea de integrare**

Testarea integrarii analizeaza precoditiile inainte de a executa rutine sau metode, care depind de anumiti parametri realizati prin cod, si se incadreaza la stategia testarii incapsulate.

**Testarea securităţii**

Modalitatea de testare va ţine cont de specificul aplicaţiei. Se va testa dacă produsul protejează confidenţialitatea, integritatea şi disponibilitatea datelor. Evaluarea gradului de ameninţare se va face după modelul DREAD, care constă în atribuirea note de la 1 la 3 conform următoarelor 5 atribute:

* Dauna potenţială
* Reproductibilitatea ameninţării prin exploatare repetată
* Exploatabilitatea (dificultatea tehnică de exploatare a vulnerabilităţii)
* Afectarea (numărul de utilizatori afectaţi)
* Descoperirea (dificultatea descoperirii vulnerabilităţii)

Notele se însumează pentru a obţine indicele DREAD. Se va fixa un prag de relevanţă sub care ameninţările vor fi ignorate de manager.

**Testarea pe bază de cerințe de business**

În funcție de specificațiile/cerințele ce se regăsesc în capitolul 1.4 se vor realiza teste prin care se va urmări dacă sunt identificate erori, omisiuni, ambiguități sau limite.

**Obiective**: Se va verifica faptul că instalarea aplicaţiei se realizează cu succes de către administratorul aplicației din cadrul hotelului, cu respectarea etapelor prevăzute în documentaţia aplicaţiei. În acelaşi timp, testerii se vor asigura că funcţionalităţile de bază, cât şi cerințele de business prevăzute sunt îndeplinite cu succes, iar interfețele aplicației sunt prietenoase cu utilizatorii.

## Elemente de testare

**Testarea instalării/dezinstalării aplicației**

* Se verifică dacă kit-ul de instalare conține fișierele cu privire la specificațiile de instalare, dar și executabilul *franco.exe*;
* Se testează finalizarea cu succes a procesului de instalare;
* Se testează dacă în START este creat grupul Franco ce cuprinde executabilul, fișierele Inceput.txt, Licenta.txt, Sfarsit.txt, manualul utilizatorului, precum și celelalte resurse necesare rulării optime a aplicației;
* Se testează dacă după instalarea aplicaţiei fişierele aplicaţiei, dar și executabilul de dezinstalare sunt prezente în fişierul stabilit de către utilizatorul care instalează;
* Se dezinstalează aplicaţia şi se verifică ştergerea fişierelor create la instalare;
* Se verifică dacă au finalizate toate procesele privind aplicația din Task Manager.

**Testarea securităţii**

* Se verifică posibilitatea logării pentru toate categoriile de utilizatori (administrator, manager, client și vizitator);
* Se verifică apariția modulelor în meniu în funcție de categoria din care face parte utilizatorul logat;
* Se verifică dacă se poate autentifica un utilizator cu o combinație greșită cont-parolă;
* Se verifică dacă se blochează un cont după trei încercări de conectare eșuate;
* Se verifică dacă la completarea parolei aceasta e ascunsă sub formă de asterisc.
* Se verifică dacă aplicația poate fi folosită de utilizatori neînregistrați.
* Se verifică apariția și funcționarea meniului pe toate categoriile de utilizatori;

**Testarea pozitivă**

* Se verifică funcționarea butoanelor prin intermediul unui click;
* Se verifică funcționalitatea câmpurilor editabile prin introducerea unor caractere în limitele constrângerilor;
* Se verifică finalizarea tranzacțiilor prin completarea datelor necesare cu unele conforme.

**Testarea negativă**

* Se verifică funcționalitatea câmpurilor editabile prin introducerea unor caractere alfa în loc de numerice și invers;
* Se verifică dacă este posibilă introducerea unei valori negative în campuri numerice;
* Se verifică dacă pot fi lăsate valori nule în campurile obligatorii;
* Se verifică dacă pot fi introduse valori cu dimensiuni mai mari decât cele conforme;
* Se verifică finalizarea tranzacțiilor prin completarea datelor necesare cu unele neconforme.

**Testarea interfeţei**

* Se verifică apariția siglei centrului pe pagina principală a aplicației;
* Se verifică opțiunile de iesire și minimizare a paginii principale a aplicației;
* Se verifică opțiunile de iesire a unui formular;
* Se verifică dacă machetele aplicației sunt prietenoase cu utilizatorii (culoare, navigare, accesibilitate, sugestivitate, inteligibilitate, specific);
* Se testează dacă în urma selectării de către utilizator a diverselor criterii de interogare raportul furnizează datele solicitate.

**Testarea valorilor limită**

* Se verifică introducerea valorilor limită în câmpurile aferente (tarif, dată rezervare, număr de săli disponibile, procentaj reducere etc.).

**Testarea de traseu**

* Verificarea traseului de realizare a unei rezervări (înregistrarea/logarea cu un cont de utilizator valid, selectarea datelor de cazare și decazare, vizualizarea tipurilor de camere disponibile, selectarea camerei dorite de utilizator, vizualizare facilități cameră selectată, selectarea opțiunii mic dejun, completarea datelor personale, selectarea metodei de plată, afișarea opțiunii alese, primire e-mail confirmare rezervare)

**Testarea pe bază de cerințe de business**

* Se verifică dacă utilizatorul este obligat să își schimbe parola la prima autentificare
* Se verifică accesul utilizatorilor la diferitele module din cadrul meniului și dacă se realizează în funcție de drepturile atribuite acestora.
* Se verifică dacă rezervările pot fi efectuate doar de către userii ce dețin un cont activ, cu statut de client.
* Se verifică ca un client să nu poată iniția o rezervare pe numele unei alte persoane (prin verificarea datelor personale de pe cardul de identitate).
* Se verifică dacă unei rezervări îi corespunde un număr unic de identificare.
* Se verifică dacă un client nu poate rezerva mai mult de 5 camere în cadrul aceleași rezervări.
* Se verifică dacă o rezervare poate fi anulată cel târziu cu 3 zile înainte de data de început a sejurului.
* Se verifică dacă o rezervare nu poate fi efectuată pentru mai puțin de o noapte.
* Se verifică dacă utilizatorul va fi delogat automat în situația în care nu este activ timp de o oră.
* Se verifică dacă fiecare raport conține perioada analizată, dar și data la care acesta a fost generat.
* Se verifică dacă data generării raportului este preluată automat din sistem.
* Se verifică dacă numărul zilelor de cazare pentru camere se calculează automat în funcție de data de început și data de final a sejurului.
* Se verifică dacă valoarea totală a unei rezervări este calculată automat în funcție de numărul de zile din sejur și tariful serviciilor prestate.
* Se verifică dacă valoarea închirierii unei săli (restaurant) poate fi negociată sau poate fi preluată din lista tarifelor standard.
* Se verifică dacă în modulele Rezervări și Săli conferințe hotelul dispune de 55 de camere (15 single și 40 duble) și de 5 săli de conferințe.
* Se verifică dacă camerele deja rezervate nu vor apărea în lista camerelor disponibile.
* Se verifică dacă ștergerea recenziilor de la clienți nu este posibilă.
* Se verifică dacă acordarea depturilor utilizatorilor se face doar de către administrator.

## Caracteristici ce trebuie testate

Se vor lua în considerare următoarele aspecte de testat:

1. Management acces/părăsire aplicaţie;
2. Management acces site utilizator;
3. Management homepage si meniu;
4. Management camere;
5. Management sali de conferinte;
6. Management clienți;
7. Managementul rapoartelor privind rezervarile;
8. Managementul rapoartelor privind clienții;
9. Managementul accesului la secțiunea de suport (*Help*).

## Caracteristici ce nu trebuie testate

Funcţionalităţile ce nu fac obiectul prezentului plan de testare sunt:

* Lucru în rețea și nici în alte medii integrate;
* Testarea de guerila;
* Respectarea normelor juridice;
* Confirmarea contului user-ului pe e-mail;
* Verificare captcha ”I’m not a robot” la autentificare;

În ceea ce privește specificațiile legate de CONT

* Resetare parolă după 3 încercări eșuate;
* Recuperare parolă;
* *Completarea datelor personale de către utilizator*
* Selectarea metodei de plată a rezervării

În ceea ce privește specificațiile legate de REZERVĂRI

* Trimitere e-mail cod promoțional utilizatorilor fideli cu minim rezervări
* Completarea datelor personale de către utilizator
* Selectarea metodei de plată
* Oferire ajutor prin intermediul chatului (În ceea ce privește specificațiile legate de AJUTOR)
* Frecvența utilizării serviciilor (Utilizatorul/Recepționerul poate vedea care sunt cele mai solicitate sercivii în cadrul hotelului) (În ceea ce privește specificațiile legate de RAPOARTE)
* Creare reduceri (pentru anumite perioade) (În ceea ce privește specificațiile legate de GESTIUNEA SERVICIILOR)

## Abordare

Pentru aplicația realizată am utilizat arhitecturile: Top-down și Bottom-up.

Pentru o mai bună verificare a integrării modulelor și reducerea timpului necesar desfășurării activități, am folosit strategia *Top-down*. Aceasta presupune începerea cu modulul de pe nivelul cel mai înalt și continuă cu următorul modul care trebuie să fie unul din modulele subordonate unui modul deja testat. Totodată, metoda Top-down constituie una din cele mai prolifice strategii de testare, în cazul testării caracteristicilor unei aplicaţii.

Avantajele pe care strategia Top-down le oferă sunt:

- Logica și flowurile principale de date sunt testate mai devreme, deci orice greșeala de design este depistată în timp util;

- Odată ce funcțiile I/O sunt adăugate, reprezentarea cazurilor de testare este mai ușoară.

- Schema timpurie a scheletului permite o mai bună înțelegere a aplicației și ușurează munca depusă de echipa de testare.

Pentru testarea aplicaţiei se va recurge la testarea manuală. Sarcinile de testare vor fi distribuite de către team leader-ul fiecărui membru a echipei de testare.

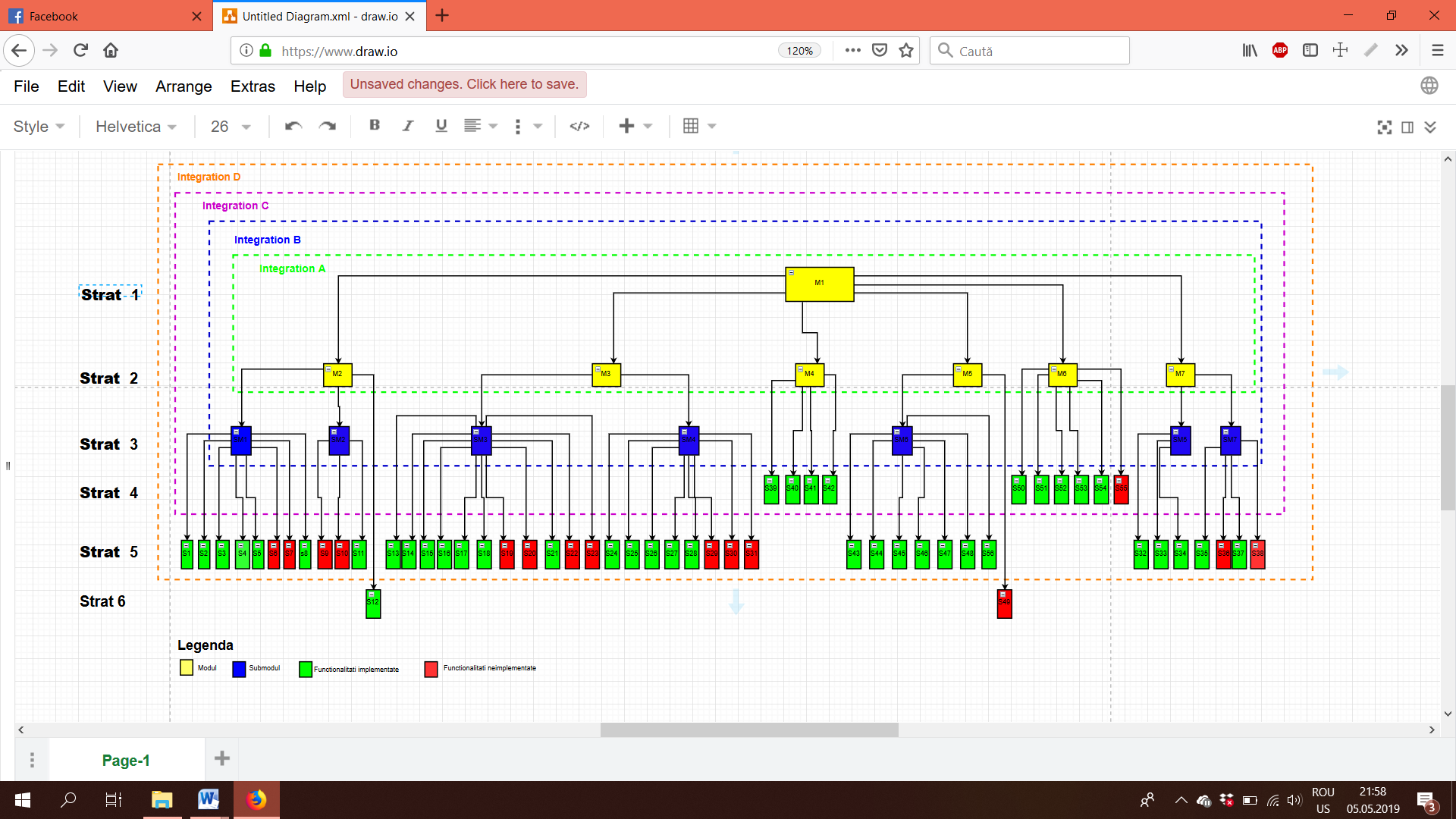


Figura 2.7.1 Arhitectura module – testare Top-down

*Explicatii integrare Top - Down*

Stage 1 – Testare unitară a modulului M1

Stage 2 – Test de integritate A a modulului M1 cu modulele M2, M3, M4, M5, M6 si M7, dezvoltate în totalitate sau parțial (aflate în starea curentă)

Stage 3 – Test de integritate B a lui A cu submodulelele SM1, SM2, SM3, SM4, SM5, SM6, SM7

Stage 4 – Test de integritate C a lui B cu S39, S40, S41, S42, S50, s51, S52, S53, S54, S55

Stage 5 – Test de integritate D a lui C cu S1...S11; S13... S23; S24....S31; S43...S48, S56; S32... S38

Stage 6 – Test de sistem

A doua strategie folosită pentru testarea aplicației este Bottom-up. Aceasta presupune testarea componentelor de la nivelul cel mai jos, urcând progresiv către componentele de nivel înalt către ansamblul sistemului. La nivelul cel mai de jos, se integrează specificațiile stabilite în modulele respective, apoi sunt testate.

Avantajele pe care strategia Bottom-up le oferă sunt:

- tip de testare este foarte eficace în găsirea bug-urilor / erorilor

- ușor de monitorizat (se poate ști în fiecare moment ce procent din teste a fost terminat)

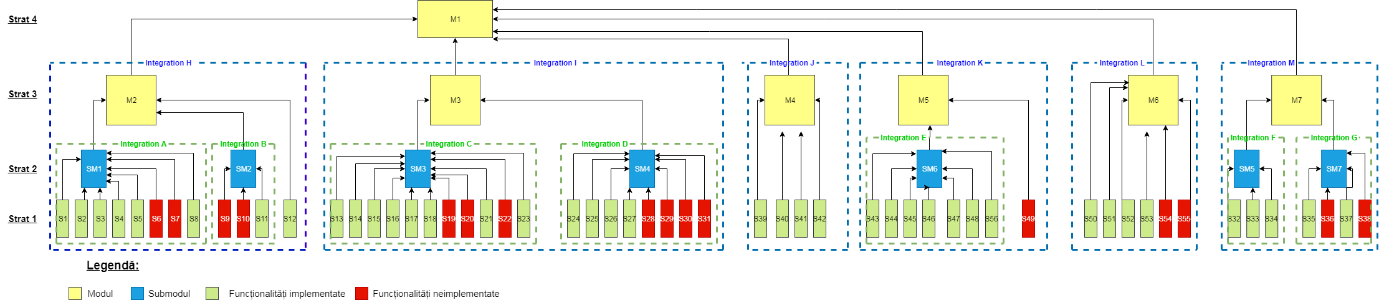


Figura 2.7.2 Arhitectura module – testare Botom-Up

*Explicatii integrare Botom - Up*

Stage 1- Testare unitară de la S1 la S56

Stage 2- Test de integritate A, B, C, D, E, F, G a submodulelor SM1, SM2, SM3, SM4, SM5, SM6, SM7 cu S1...S11; S13...S31; S43...S48; S56; S32...S34; S35...S38

Stage 3- Test de integritate H, I, J, K, L, M a modulelor M2, M3, M4, M5, M6, M7 cu SM1...SM7 și S1...S56

Stage 4- Test de sistem a modulului M1

## Criterii de trecere/picare a testului de către elementele testate

**Criteriile de trecere a testului de către elementele testate se referă la**:

a. Instalarea cu succes a aplicaţiei prin rularea kit-ului (*franco.exe*);

b. Verificarea lansării cu succes în execuţie a aplicaţiei (prin intermediul bookmark-ului *HotelFranco* din browser);

c. Existența secțiunii de logare în fereastra principală a aplicației;

d. Verificarea posibilității de logare în cadrul aplicației în funcție de categoriile de utilizatori ( 4 categorii de useri: administrator, manager, client și vizitator);

e. Afișarea meniului principal și a submeniurilor conform nivelului de acces al categoriei din care face parte utilizatorul logat;

f. Accesarea formularelor și rapoartelor conform nivelului de acces al categoriei din care face parte utilizatorul logat;

**Criteriile de ieşire din efectuare a testelor se referă la**:

a. Oricând se constată o încălcare a cerințelor de business;

b. Oricând se constată o eroare fatală;

c. Realizarea tuturor cazurilor de testare stabilite pentru aplicaţie;

d. Epuizarea timpului alocat testării;

e. Oricând este justificat, prin aprobarea leader-ului echipei de testare.

**Criteriile de continuare a testelor:**

1. După fiecare refacere a aplicaţiei;
2. Oricând este justificat, prin aprobarea leader-ului echipei de testare.

## Criteriile de suspendare/reluare a procesului de testare

Decizia de suspendare a testelor va fi condiţionată de neîndeplinirea uneia dintre funcţionalităţile de bază ale aplicaţiei definite în specificaţiile sistemului:

• imposibilitatea accesării meniului și a formularelor;

• imposibilitatea realizării de noi tranzacții, a rezervarilor, inregistrare user nou, serviciul nou (administrator);

• imposibilitatea vizualizării rapoartelor, formularelor, salilor / camere libere,.

Reluarea procesului de testare va fi făcută doar după soluţionarea problemei de către dezvoltatori.

## Rezultate finale ale testelor

În scopul raportării către echipa de dezvoltare a rezultatelor finale a testelor efectuate de către echipa de testare se va realiza un raport de testare ce va conține:

* descrierea fiecărei erori;
* descrierea bug-urilor;
* prezentarea cazurilor de testare, a parametrilor utilizați, a rezultatului așteptat și a celui obținut.

## Sarcini de testare

Sarcinile în vederea testarii aplicaţiei vor fi alocate de către team leader, urmând a fi îndeplinite de fiecare membru din echipa de testare ce are drept atribuții definirea și executarea cazurilor de test, urmărind documentarea acestora într-un format comun, tot procesul finalizându-se cu realizarea raportului de testare.

Sarcinile de testare sunt următoarele:

* Realizare cazuri de testare;
* Execuţie teste;
* Raportare;
* Rezolvare erori;
* Testare regresivă;
* Test de acceptanţă la beneficiar.

## Nevoi de mediu

- dispozitive cu procesor Intel sau AMD, configurație cu minimum 1G spațiu liber și minimum 1G memorie RAM;

- nu sunt necesare drepturi de ***administrator*** pentru instalarea și utilizarea aplicației;

- dispozitivele trebuie să ruleze pe sisteme de operare Windows în orice versiune Windows disponibilă;

- sisteme cu Google Chrome instalat;

- sisteme cu Microsoft SQL Server 2017, Microsoft ODBC Driver 17 for SQL Server și .NET Core SDK 2.2 (sau o versiune mai recentă) instalate;

## Responsabilităţi

Modul în care sarcinile aferente activității de testare au fost repartizare membrilor echiper este prezentat în tabelul de mai jos:

**Tabel 2.1 - Tabel cu repartizarea sarcinilor de testare**

|  |  |  |
| --- | --- | --- |
| **Sarcină** | **Responsabil** | |
| **Realizare cazuri de testare** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |
| **Execuţie teste** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |
| **Raportare** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |
| **Rezolvare erori** | Petrescu Petre  Perte Ardela  Lambrino Mirela-Viorica (frontend) | |
| **Testare regresivă** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |
| **Test de acceptanţă la beneficiar** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |

## Nevoi de personal şi specializare

Echipa responsabilă de realizarea activitații de testare este formată din 7 membri dintre care 2 au rolul și de proiectanți/dezvoltatori.

## Program

**Tabel 2.2 - Tabel cu programul sarcinilor de testare**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sarcină** | **Responsabil** | | **Termen** |
| **Realizare cazuri de testare** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |  |
| **Execuţie teste** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |  |
| **Raportare** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |  |
| **Rezolvare erori** | Petre Petrescu  Perte Ardela  Lambrino Mirela-Viorica | | În funcție de problemă |
| **Testare regresivă** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela - Viorica |  |
| **Test de acceptanţă la beneficiar** | PloaieArdela  Barjă Ana - Mădălina  Parlung Valy  Perte Ardela | Vrabie Oana  Stanca Clara - Ardela  Lambrino Mirela – Viorica |  |

## Riscuri şi evenimente neprevăzute

- repartizarea unor membri din echipa de testare pe alte proiecte, fapt ce ar duce la neacoperirea nevoilor de personal pe proiectul prezent și astfel, la neîncadrarea în timpul stabilit; lipsa de personal, chiar și temporară, ar putea conduce, de altfel, și la scăderea calității produsul software livrat;

- bug-urile descoperite ar putea necesita un timp mai îndelungat de fixare și retestare decât a fost inițial prevăzut; de altfel, fixarea unor bug-uri ar putea conduce la crearea altora în alte părți ale aplicației;

- eventuale probleme ale furnizorului de internet ar putea conduce la întarzierea tuturor activităților necesare livrării aplicației către beneficiar la termenul stabilit; aceeași problemă poate afecta și disponibilitatea aplicației află deja în folosință la beneficiar;

- nefuncționarea aplicației datorită nerespectării indicațiilor oferite în fișierele Început.txt și Licență.txt; reticiența utilizatorilor în utilizarea aplicației, probleme de acomodare cu noul sistem software;

## Aprobări

Toate elementele planului de testare a aplicaţiei ***Franco*** au fost supuse aprobării tuturor părţilor implicate în elaborarea aplicaţiei, aşa cum se prezintă în tabelul de mai jos.

**Tabel 2.3 - Aprobări**

|  |  |  |
| --- | --- | --- |
| **Responsabil** | **Data** | **Semnătură** |
| Manager de proiect: Petrescu Petre |  |  |
| Beneficiar: SC Hotel Franco SRL |  |  |
| Analist programator: Parlung Valy |  |  |
| Manager Echipă Test:Barjă Ana - Mădălina |  |  |
| Manager Echipă Dezvoltare: Petre Petrescu |  |  |