Proiectarea și Implementarea unei Platforme Informatice pentru Gestionarea unui Magazin de Animale

# Documentul de proiectare

CUPRINS

[Documentul de proiectare i](#_Toc164106059)

[1. Introducere 1](#_Toc164106060)

[1.1 Scopul documentului 1](#_Toc164106061)

[2. Prezentare generală și abordări de proiectare 3](#_Toc164106062)

[2.1 Prezentare generală 3](#_Toc164106063)

[2.2 Presupuneri/ Constrângeri/ Riscuri 4](#_Toc164106064)

[3. Considerațiii de proiectare 5](#_Toc164106065)

[3.1 Obiective și linii directoare (ghiduri) 5](#_Toc164106066)

[3.2 Metode de dezvoltare 5](#_Toc164106067)

[3.3 Strategii de arhitectură 5](#_Toc164106068)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 7](#_Toc164106069)

[4.1 Vedere logică 7](#_Toc164106070)

[4.2 Arhitectură hardware 8](#_Toc164106071)

[4.3 Arhitectură software 8](#_Toc164106072)

[4.4 Arhitectura informațiilor 9](#_Toc164106073)

[4.5 Arhitectura de comunicații interne 9](#_Toc164106074)

[5. Proiectarea sistemului 10](#_Toc164106075)

[5.1 Proiectarea bazei de date 10](#_Toc164106076)

[5.2 Conversii de date 13](#_Toc164106077)

[5.3 Interfețe utilizator 13](#_Toc164106078)

[5.4 Proiectarea interfețelor cu utilizatorul 13](#_Toc164106079)

[6. Scenarii de utilizare 14](#_Toc164106080)

[7. Controale pentru verificarea integrității sistemului 16](#_Toc164106081)

## Introducere

Numele Proiectului: Proiectarea și Implementarea unei Platforme Informatice pentru Gestionarea unui Magazin de Animale

Acronim: PIPIGMA

Situația Existentă: În industria comerțului cu animale, există o nevoie crescută de soluții informatice specializate pentru gestionarea eficientă a activităților magazinelor de animale. Absența unui sistem dedicat poate duce la dificultăți în gestionarea stocului, interacțiunea cu clienții și monitorizarea vânzărilor.

Situația Propusă: Proiectarea și Implementarea unei Platforme Informatice pentru Gestionarea unui Magazin de Animale reprezintă o soluție digitală propusă pentru a răspunde nevoilor magazinelor de animale de a avea o platformă eficientă și personalizată pentru gestionarea activităților lor. PIPIGMA va oferi un set de instrumente și funcționalități adaptate specificului industriei, ușurând gestionarea stocului, interacțiunea cu clienții și monitorizarea performanței magazinului.

Evoluția Așteptată a Documentului: Documentul de Proiectare a Sistemului va evolua în concordanță cu progresul proiectului. În etapele incipiente, se va concentra pe prezentarea generală a arhitecturii și obiectivelor sistemului. Pe măsură ce proiectul avansează, se vor detalia specificațiile tehnice și operaționale, precum și proiectarea detaliată a componentelor sistemului.

Considerații Referitoare la Securitate și Confidențialitate: Documentul de Proiectare a Sistemului va include o secțiune dedicată considerațiilor referitoare la securitate și confidențialitate. Se vor identifica și evalua riscurile potențiale asociate cu gestionarea datelor sensibile ale clienților și ale magazinelor de animale. De asemenea, se vor propune și implementa măsuri de securitate pentru protejarea acestor date împotriva accesului neautorizat și a amenințărilor cibernetice.

### Scopul documentului

Scopul acestui Document de Proiectare a Sistemului (SDD) este de a furniza o structură și un cadru pentru proiectarea și implementarea unei platforme informatice pentru gestionarea unui magazin de animale. Documentul va descrie arhitectura sistemului propus, principiile de proiectare și strategiile utilizate, precum și presupozițiile, constrângerile și riscurile asociate.

Prin intermediul SDD-ului, se urmărește să:

* Definim și să descriem în detaliu arhitectura sistemului propus, inclusiv componentele sale, interacțiunile între acestea și fluxul datelor în cadrul sistemului.
* Specificăm cerințele funcționale și non-funcționale ale sistemului, precum și modul în care acestea sunt îndeplinite de către proiectarea propusă.
* Identificăm și să evaluăm riscurile potențiale asociate cu arhitectura și designul sistemului, furnizând strategii și soluții pentru gestionarea acestora.
* Furnizăm detalii tehnice și operaționale pentru implementarea efectivă a sistemului, inclusiv structura datelor, interfața utilizatorului și aspectele de securitate și confidențialitate.
* Documentăm modificările și iterațiile ulterioare ale proiectării pe parcursul ciclului de viață al dezvoltării sistemului, pentru a reflecta schimbările și adaptările necesare în timpul evoluției proiectului.

## Prezentare generală și abordări de proiectare

În această secțiune, vom prezenta principiile și strategiile pe care le vom utiliza ca ghiduri în proiectarea și implementarea sistemului "Proiectarea și Implementarea unei Platforme Informatice pentru Gestionarea unui Magazin de Animale" (PIPIGMA).

* **Flexibilitate și scalabilitate:** Un principiu fundamental al proiectului nostru este de a proiecta o platformă flexibilă și scalabilă, capabilă să se adapteze la nevoile și cerințele în continuă schimbare ale magazinelor de animale. Acest lucru va fi realizat prin adoptarea unei arhitecturi modularizate și a unei structuri flexibile de date, care să permită adăugarea și modificarea funcționalităților fără a afecta întregul sistem.
* **Experiența utilizatorului:** O atenție deosebită va fi acordată proiectării unei interfețe cu utilizatorul (UI) intuitive și ușor de utilizat. Vom prioritiza experiența utilizatorului, facilitând navigarea fluidă și accesul rapid la informații și funcționalități relevante. Interfața utilizatorului va fi proiectată pentru a fi consistentă, estetică și adaptată nevoilor specifice ale utilizatorilor din industria magazinelor de animale.
* **Securitate și confidențialitate:** Un aspect crucial al proiectului nostru este asigurarea securității și confidențialității datelor. Vom implementa măsuri de securitate robuste pentru protejarea informațiilor sensibile ale clienților și ale magazinelor de animale. Aceasta include autentificare sigură, autorizare adecvată și criptare a datelor sensibile în tranzit și la repaus.
* **Performanță și eficiență:** Ne propunem să proiectăm și să implementăm un sistem cu performanțe ridicate și eficiență optimă. Vom optimiza algoritmul și baza de date pentru a asigura timpuri de răspuns scurte și o gestionare eficientă a resurselor. De asemenea, vom lua în considerare posibilitatea de a utiliza tehnologii moderne pentru a maximiza eficiența operațională a sistemului.
* **Adaptabilitate și reutilizabilitate:** Vom adopta o abordare modulară în proiectarea sistemului, permițând componentelor să fie dezvoltate și testate independent. Aceasta va facilita reutilizarea și extensibilitatea codului, precum și adaptabilitatea sistemului la cerințele viitoare și evoluția tehnologiei.

### Prezentare generală

Platforma informatică propusă este concepută pentru a gestiona toate aspectele operaționale ale unui magazin de animale, inclusiv stocurile de animale și produse, gestionarea clienților, programarea serviciilor veterinare, generarea de rapoarte și analize, și gestionarea financiară. Arhitectura sistemului constă într-o aplicație web cu interfețe intuitive pentru utilizatori și o bază de date centrală pentru stocuri, clienți și tranzacții financiare. Obiectivul principal al proiectării este de a oferi o soluție scalabilă, sigură și ușor de utilizat pentru gestionarea eficientă a magazinului de animale.

### Presupuneri/ Constrângeri/ Riscuri

#### Presupuneri

Utilizatorii finali au acces la un dispozitiv cu conexiune la internet pentru a accesa platforma.

Personalul magazinului va fi instruit în utilizarea sistemului.

Există un nivel minim de cunoștințe tehnice necesare pentru utilizarea platformei.

#### Constrângeri

Limitările hardware și software ale dispozitivelor utilizate de personalul magazinului pot influența performanța sistemului.

Respectarea standardelor de securitate și confidențialitate a datelor cu caracter personal ale clienților și informațiilor financiare este obligatorie conform reglementărilor legale.

#### Riscuri

Unul dintre riscurile majore este reprezentat de posibilele atacuri cibernetice asupra sistemului, motiv pentru care vor fi implementate măsuri stricte de securitate, cum ar fi criptarea datelor și autentificarea în doi factori.

Riscul de indisponibilitate a sistemului din cauza eșecurilor hardware sau software va fi gestionat prin implementarea unui plan de backup și de recuperare a datelor.

## Considerațiii de proiectare

Definirea clară a cerințelor: Este crucial să înțelegem în detaliu cerințele sistemului și a nevoilor utilizatorilor pentru a putea proiecta o soluție adecvată.

Evaluarea tehnologiilor disponibile: Trebuie să analizăm și să evaluăm tehnologiile disponibile pentru a alege cele mai potrivite pentru implementarea sistemului PIIGMA.

Identificarea și gestionarea riscurilor: Trebuie să identificăm și să evaluăm riscurile asociate cu proiectul, inclusiv riscurile legate de securitate, performanță și scalabilitate.

### Obiective și linii directoare (ghiduri)

Principalele obiective ale proiectului includ:

* + *Eficiența operațională*: Sistemul trebuie să ofere o gestionare eficientă a stocurilor de animale și produse, programarea serviciilor veterinare și gestionarea financiară.
  + *Ușurința în utilizare*: Interfețele trebuie să fie intuitive și ușor de înțeles pentru utilizatorii finali.
  + *Siguranța și securitatea datelor*: Este esențial să se asigure protecția datelor cu caracter personal ale clienților și informațiilor financiare.
  + *Scalabilitatea*: Sistemul trebuie să fie proiectat pentru a putea crește în funcționalitate și capacitate odată cu extinderea magazinului de animale.
  + *Flexibilitatea*: Arhitectura sistemului ar trebui să permită adăugarea ușoară a noi funcționalități sau integrarea cu alte sisteme.

### Metode de dezvoltare

Dezvoltarea sistemului se va baza pe o abordare orientată pe obiecte, utilizând tehnologii moderne de dezvoltare web. Se va folosi o combinație de tehnologii frontend și backend, cum ar fi HTML, CSS și JavaScript pentru interfața utilizatorului și framework-uri și limbaje de programare backend cum ar fi PHP sau SQL. Designul sistemului va fi ghidat de principiile de inginerie software și de bune practici de programare.

### Strategii de arhitectură

Principalele decizii și strategii de design care afectează organizarea generală a sistemului PIPIGMA includ:

* Utilizarea unui limbaj de programare orientat pe obiecte, cum ar fi Java, pentru implementarea componentelor software.
* Reutilizarea componentelor software existente, atunci când este posibil, pentru a minimiza efortul de dezvoltare și pentru a maximiza eficiența sistemului.
* Planificarea extinderii și îmbunătățirii ulterioare a sistemului prin adoptarea unei arhitecturi modularizate și a unor standarde deschise pentru interoperabilitate.
* Implementarea unei politici robuste de gestionare a erorilor și a mecanismelor de recuperare pentru a asigura fiabilitatea și stabilitatea sistemului.
* Utilizarea unei baze de date relaționale pentru gestionarea și persistența datelor, cu implementarea unor măsuri adecvate de securitate și performanță.
* Abordarea concurenței și sincronizării în mod corespunzător pentru a asigura consistența și integritatea datelor în medii distribuite și concurente.

Comunicarea între componentele sistemului va folosi protocoale standard și se va implementa gestionarea tranzacțiilor și a concurenței.

## Arhitectura Sistemului și Proiectarea Arhitecturii

Arhitectura sistemului PIPIGMA este concepută pentru a fi modulară și scalabilă, astfel încât să poată gestiona eficient cerințele unei platforme web pentru gestionarea unui magazin de animale. Sistemul este format din următoarele componente/subsisteme:

* **Interfața cu Utilizatorul (UI)**: Aceasta este componenta de nivel superior care se ocupă cu interacțiunea directă cu utilizatorii sistemului. UI-ul furnizează o interfață web intuitivă și prietenoasă pentru utilizatori, permițându-le să caute și să achiziționeze produse, să gestioneze conturile lor și să efectueze plăți.
* **Logica de Afaceri**: Acest subsistem conține logica de afaceri a platformei PIIGMA. Este responsabil pentru procesarea comenzilor, gestionarea stocului de produse, calcularea prețurilor și aplicarea regulilor de afaceri. Acesta asigură integritatea și consistența datelor din sistem și implementează funcționalități complexe precum gestionarea promoțiilor și a ofertelor.
* **Gestionarea Datelor**: Acest subsistem gestionează persistența datelor utilizate de sistem. Include baza de date și serviciile asociate care se ocupă cu stocarea, recuperarea și actualizarea informațiilor despre produse, clienți, comenzi etc. Gestionarea datelor este proiectată pentru a asigura performanță și fiabilitate în manipularea volumelor mari de date.
* **Integrarea cu Serviciile Externe**: Această componentă se ocupă cu integrarea sistemului PIIGMA cu alte servicii externe, cum ar fi sistemele de plată online sau serviciile de livrare. Ea facilitează schimbul de date și interacțiunile cu terțe părți, permițând sistemului să ofere funcționalități extinse și să se integreze în ecosistemele existente.

### Vedere logică

Arhitectura sistemului PIPIGMA este concepută pentru a îndeplini cerințele funcționale și non-funcționale ale unei platforme informatice pentru gestionarea unui magazin de animale. Componentele sistemului sunt organizate într-o manieră modulară, asigurând astfel o dezvoltare eficientă și scalabilă. Prin intermediul unei abordări orientate pe obiecte și a unor tipare de proiectare bine definite, interacțiunile dintre aceste componente sunt gestionate eficient.

În cadrul PIPIGMA, vederea logică ar putea include:

* **Structura Bazei de Date**: Aceasta ar putea include modelele entitate-relație (ER) sau orice altă formă de structură a bazei de date utilizate pentru stocarea datelor sistemului, cum ar fi tabelele și relațiile între acestea.
* **Logică de Procesare**: Acesta este codul și logica care manipulează datele în cadrul aplicației. Aici sunt incluse procesele de procesare a comenzilor, calculul prețurilor, gestionarea stocului etc.
* **Fluxurile de Informații**: Acest aspect se referă la modul în care datele sunt transmise și procesate în cadrul aplicației, inclusiv fluxurile de date între diferitele componente și subsisteme.

### Arhitectură hardware

Sistemul PIPIGMA este implementat pe o infrastructură hardware distribuită, care include:

* **Servere de prezentare**: Acestea sunt servere care găzduiesc interfața cu utilizatorul (UI) a platformei web PIPIGMA și furnizează funcționalitățile asociate cu partea de client a aplicației.
* **Servere de aplicație**: Aceste servere găzduiesc logica de afaceri și funcționalitățile de gestionare a datelor ale sistemului PIPIGMA.
* **Servere de date**: Aceste servere stochează și gestionează datele utilizate de sistem, inclusiv informațiile despre produse, clienți și comenzi.
* **Dispozitive periferice**: Acestea includ dispozitive precum load balancers pentru distribuirea traficului și firewall-uri pentru securizarea infrastructurii.

Arhitectura hardware a sistemului este concepută pentru a asigura performanță, scalabilitate și disponibilitate, fiind configurată pentru a satisface cerințele operaționale ale platformei PIPIGMA.

### Arhitectură software

Arhitectura software a sistemului PIPIGMA este formată dintr-o serie de componente și module care colaborează pentru a oferi funcționalitățile dorite. Acestea includ:

* Componente de prezentare: Interfața cu utilizatorul (UI) este implementată folosind tehnologii web moderne, precum HTML, CSS și JavaScript. Acest strat de prezentare este responsabil pentru afișarea datelor și interacțiunea cu utilizatorul final.
* Componente de aplicație: Logica de afaceri a sistemului este implementată în aceste componente, folosind limbaje de programare precum Java. Aceste componente gestionează operațiile de gestionare a stocului, procesarea comenzilor și interacțiunea cu baza de date.
* Componente de date: Accesul la date este gestionat de aceste componente, care includ biblioteci de acces la baze de date și module de gestionare a datelor. Aceste componente sunt responsabile pentru interacțiunea cu baza de date și pentru manipularea datelor.
* Servicii externe: Sistemul PIPIGMA poate interacționa cu alte servicii externe, cum ar fi servicii de plată online sau sisteme de livrare. Aceste interacțiuni sunt gestionate de componente specifice, care se integrează cu API-uri externe.

Arhitectura software este proiectată pentru a fi modulară, cu componente independente care pot fi dezvoltate, testate și întreținute separat. Această abordare permite o dezvoltare eficientă și o gestionare ușoară a sistemului PIPIGMA.

### Arhitectura informațiilor

Informațiile stocate în sistemul PIPIGMA includ detalii despre produse, clienți, comenzi și alte entități relevante pentru gestionarea magazinelor de animale. Aceste date sunt gestionate într-o bază de date relațională, care este proiectată pentru a asigura integritatea și confidențialitatea informațiilor.

### Arhitectura de comunicații interne

Comunicațiile interne în cadrul sistemului PIPIGMA sunt gestionate printr-o rețea internă care conectează toate componentele și subsistemele. Această rețea este configurată pentru a asigura o comunicare rapidă și sigură între diferitele elemente ale sistemului, inclusiv serverele de prezentare, aplicație și date.

## Proiectarea sistemului

### Proiectarea bazei de date

**1. Tabelul "Clienti"**

* ID\_Client (int, cheie primară, auto-incremențare): Identificator unic al clientului
* Nume\_Client (varchar(50)): Numele clientului
* Prenume\_Client (varchar(50)): Prenumele clientului
* Email (varchar(100)): Adresa de email a clientului
* Telefon (varchar(20)): Numărul de telefon al clientului
* Adresa (varchar(100)): Adresa clientului

**2. Tabelul "Comenzi"**

* ID\_Comanda (int, cheie primară, auto-incremențare): Identificator unic al comenzii
* ID\_Client (int, cheie externă): Cheia externă către tabelul "Clienti" pentru a identifica clientul care a plasat comanda
* Data\_Comanda (datetime): Data și ora la care a fost plasată comanda

**3. Tabelul "Categorii"**

* ID\_Categorie (int, cheie primară, auto-incremențare): Identificator unic al categoriei
* Nume\_Categorie (varchar(50)): Numele categoriei (ex: câini, pisici, păsări)

**4. Tabelul "Plati"**

* ID\_Plata (int, cheie primară, auto-incremențare): Identificator unic al plății
* ID\_Comanda (int, cheie externă): Cheia externă către tabelul "Comenzi" pentru a identifica comanda asociată plății
* Suma\_Plata (float): Suma plătită
* Data\_Plata (datetime): Data și ora plății

**5. Tabelul "Vanzari"**

* ID\_Vanzare (int, cheie primară, auto-incremențare): Identificator unic al vânzării
* ID\_Comanda (int, cheie externă): Cheia externă către tabelul "Comenzi" pentru a identifica comanda asociată vânzării
* Data\_Vanzare (datetime): Data și ora vânzării

**6. Tabelul "Animale"**

* ID\_Animal (int, cheie primară, auto-incremențare): Identificator unic al animalului
* Nume\_Animal (varchar(50)): Numele animalului
* ID\_Categorie (int, cheie externă): Cheia externă către tabelul "Categorii" pentru a identifica categoria animalului
* Rasă (varchar(50)): Rasa animalului
* Data\_Nastere (date): Data nașterii animalului
* Pret (float): Prețul de vânzare al animalului
* Disponibilitate (bit): Indicator pentru disponibilitatea animalului în stoc (0 = indisponibil, 1 = disponibil)

#### Obiecte de date și structuri de date rezultante

Pentru fiecare obiect de date funcțional, cum ar fi animalele, produsele, clienții și programările, se vor specifica structurile de date corespunzătoare pentru stocarea și procesarea informațiilor relevante. De exemplu, structura de date pentru animal ar putea include câmpuri precum numele, specia, rasă, vârsta, sexul etc.

#### Fișiere și baze de date

A screenshot of a computer

Description automatically generated

Figura 1. Baza de date

##### Baze de date

Proiectarea detaliată a bazei de date va include definirea tabelelor pentru fiecare entitate din sistem, împreună cu relațiile între acestea. De exemplu, o bază de date ar putea include tabele pentru animale, produse, clienți, programări etc., fiecare cu câmpurile lor specifice.

##### Fișiere non-DBMS

Fișierele non-DBMS vor fi utilizate pentru scopuri precum stocarea de imagini ale animalelor, a facturilor sau a altor documente asociate cu magazinul. Aceste fișiere vor fi utilizate pentru intrare, ieșire sau ambele, în funcție de necesități. Se vor furniza detalii despre structura și utilizarea fiecărui fișier, precum și despre modul în care acestea sunt gestionate de către sistem.

**Fișierul de configurare:**

Nume: config.txt

Descriere: Acest fișier conține setările de configurare ale aplicației, cum ar fi adresa serverului de bază de date, portul de conectare, credențialele de acces etc.

Utilizare: Fișierul este utilizat pentru intrare de către modulul de configurare al sistemului pentru a obține setările necesare pentru conectarea la baza de date.

**Fișierul de jurnalizare (log):**

Nume: system\_log.txt

Descriere: Acest fișier este utilizat pentru a înregistra evenimente importante sau erori care apar în timpul funcționării sistemului.

Utilizare: Fișierul este utilizat pentru ieșire pentru a înregistra informații despre activitatea sistemului, cum ar fi mesajele de eroare, avertismente și informații de depanare.

**Fișierul de rapoarte:**

Nume: reports.csv

Descriere: Acest fișier conține rapoartele generate de sistem, cum ar fi rapoarte de vânzări, rapoarte de stoc etc.

Utilizare: Fișierul este utilizat pentru ieșire pentru a furniza rapoarte pentru analiza datelor și raportarea performanței magazinului.

**Fișierul de backup:**

Nume: backup\_data.zip

Descriere: Acest fișier conține o copie de rezervă a datelor importante ale sistemului, cum ar fi baza de date, configurațiile și alte fișiere critice.

Utilizare: Fișierul poate fi utilizat pentru ambele, intrare și ieșire, pentru a realiza backup-uri periodice ale datelor și pentru a restaura sistemul în caz de pierdere sau corupție a datelor.

### Conversii de date

Orice conversii de date necesare pentru a migra datele existente sau pentru a integra datele provenite din surse externe vor fi documentate în detaliu, inclusiv procedurile și instrumentele utilizate pentru a asigura integritatea și consistența datelor.

### Interfețe utilizator

Pentru fiecare clasă de utilizatori, precum administratori, angajați și clienți, se va furniza o descriere a modului în care aceștia interacționează cu sistemul. Acest lucru va include mijloacele de intrare, cum ar fi ecranele de introducere a datelor, și ieșirile, cum ar fi rapoartele sau interfețele grafice de utilizator. Estimările privind numărul de utilizatori și cerințele de securitate vor fi, de asemenea, incluse.

### Proiectarea interfețelor cu utilizatorul

Interfețele cu utilizatorul vor fi proiectate pentru a oferi o experiență intuitivă și eficientă utilizatorilor sistemului. Acestea vor include ecrane de introducere a datelor, ecrane de afișare a informațiilor, meniuri și alte elemente de navigare, toate proiectate pentru a facilita interacțiunea utilizatorilor cu sistemul.

## Scenarii de utilizare

Pentru un sistem de gestionare a unui magazin de animale, iată câteva scenarii de utilizare care acoperă diverse funcționalități și interacțiuni cu utilizatorii:

**Scenariu 1: Procesul de cumpărare a unui animal**

* Utilizatorul navighează prin catalogul de animale disponibile în magazin.
* Utilizatorul selectează animalul dorit și vizualizează detaliile acestuia, precum specia, vârsta, prețul etc.
* Utilizatorul adaugă animalul în coșul de cumpărături și finalizează comanda.
* Sistemul procesează comanda, verifică disponibilitatea animalului și calculează totalul comenzii.
* Utilizatorul introduce informațiile de livrare și de plată.
* Utilizatorul finalizează achiziția, iar sistemul generează o confirmare a comenzii și trimite-o către utilizator pe email.
* Personalul magazinului pregătește animalul pentru livrare sau ridicare.

**Scenariu 2: Înregistrarea unui nou animal în sistem**

* Personalul magazinului primește un nou animal în magazin.
* Angajatul introduce detaliile animalului în sistem, precum specia, vârsta, greutatea, prețul etc.
* Sistemul generează automat un cod unic de identificare pentru animal și actualizează stocul magazinului.
* Personalul magazinului etichetează animalul și îl plasează în incinta magazinului pentru expunere.

**Scenariu 3: Gestionarea stocului de produse**

* Managerul magazinului accesează panoul de control al stocului.
* Managerul vizualizează lista de produse disponibile și stocul actual al fiecărui produs.
* Dacă un anumit produs este în pericol să se epuizeze, managerul poate plasa automat o comandă pentru a reînnoi stocul.
* Sistemul generează o comandă către furnizor și actualizează stocul în conformitate.

**Scenariu 4: Programarea unei consultații veterinar**

* Utilizatorul accesează secțiunea de programare a consultațiilor pe site-ul magazinului.
* Utilizatorul selectează data și ora dorite pentru consultație și introduce detaliile animalului.
* Sistemul verifică disponibilitatea veterinarului și alocă slotul de timp solicitat.
* Utilizatorul primește confirmarea programării pe email și poate vizualiza detaliile consultației în contul său.

Aceste scenarii acoperă aspecte esențiale ale funcționalității sistemului de gestionare a unui magazin de animale și descriu modul în care utilizatorii interacționează cu sistemul în diferite situații operaționale.

## Controale pentru verificarea integrității sistemului

**Securitate internă:**

* + Utilizarea unui sistem de autentificare robust pentru a restricționa accesul la datele critice doar pentru utilizatorii și operatorii autorizați.
  + Implementarea unui sistem de autorizare care să permită doar tipurile de acces necesare pentru fiecare utilizator sau rol în cadrul sistemului.
  + Criptarea datelor sensibile în tranzit și în repaus pentru a proteja integritatea acestora împotriva accesului neautorizat.

**Proceduri de audit:**

* + Dezvoltarea și implementarea unor proceduri de audit standardizate pentru a monitoriza și raporta activitățile sistemului.
  + Definirea unor perioade de reținere adecvate pentru rapoartele operaționale și de management conform reglementărilor sau cerințelor interne.

**Piste de auditare a aplicațiilor:**

* + Implementarea unui sistem robust de înregistrare a evenimentelor pentru a audita dinamic accesul la datele critice desemnate.
  + Asigurarea că piste de auditare sunt protejate împotriva modificărilor neautorizate sau ștergerilor.

**Tabele standard pentru validare câmpurilor de date:**

* + Definirea și utilizarea unor tabele standard pentru validarea câmpurilor de date în conformitate cu regulile și restricțiile impuse de sistem.

**Procese de verificare pentru modificarea datelor critice:**

* + Implementarea unor procese riguroase de verificare pentru a valida și autoriza adăugarea, ștergerea sau actualizarea datelor critice.
  + Atribuirea unor privilegii de acces adecvate și restricționarea accesului la funcționalitățile de modificare a datelor critice numai pentru utilizatorii autorizați.

**Capacitatea de identificare a informațiilor de auditare:**

* + Utilizarea unui sistem de înregistrare a evenimentelor care să identifice și să înregistreze toate informațiile relevante de auditare, inclusiv identitatea utilizatorului, identificarea terminalului de rețea, data și ora accesului și detaliile despre datele accesate sau modificate.

Anexa A: Gestiunea modificărilor documentului

Tabel 1 – Înregistrarea modificărilor asupreaa documentului curent

| versiune | Data | Autorul/Deținătorul | Descriere |
| --- | --- | --- | --- |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |

Anexa B: Acronime

Tabel 2 - Acronime

| Acronim | Forma completă |
| --- | --- |
| <Acronim> | <Forma completă> |
| <Acronim> | <Forma completă> |
| <Acronim> | <Forma completă> |

Anexa C Documente la care se face referire

Tabel 3 – Documente la care se facce referire

| Nume document | Locație sau URL | Dată emitere document |
| --- | --- | --- |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |