פרויקט סיום מבוא לתכנות מערכות – דו"ח פרויקט

עמרי רוטר - 206502379

אמין אסדי 322470303

**מערכת לניהול מקלט חיות מחמד**

**תיאור המערכת:**

מערכת זו משמשת כפלטפורמת ניהול מקיפה עבור מקלט לחיות מחמד. היא משלבת פונקציונליות מגוונת כדי לייעל תהליכים כגון רישום חיות מחמד, תהליכי אימוץ ומעקב אחר תרומות. המערכת בנויה סביב מספר סוגי נתונים ופעולות מרכזיים, המאפשרים קלות שימוש ויעילות בניהול צרכי המקלט.

**תיאורי מבנים מפורטים:**

1. Pet

תיאור: מבנה ה`Pet` מכיל את כל המידע הרלוונטי על חיית מחמד, כולל פרטי זיהוי, סוג, גזע וסטטוס במקלט. הוא גם מקשר לרשומות רפואיות ופרטי אימוץ כדי לספק מבט מקיף על מצבה של כל חיית מחמד.

**נתונים מוכלים:**

- מספר זהות לחיית מחמד

- שם

- סוג (לדוגמה, כלב, חתול)

- גזע

- קישור לרשומות רפואיות

- סטטוס אימוץ

**פונקציונליות:**

- רישום חיית מחמד חדשה

- עדכון פרטי חיית מחמד

- גישה ושינוי רשומות רפואיות

- בדיקה ועדכון סטטוס אימוץ

- שמירה וטעינה ממערכות אחסון

1. PetOwner

תיאור: מבנה ה`PetOwner` מייצג אנשים המאמצים חיות מחמד מהמקלט. הוא כולל פרטים אישיים וגם רשימה של חיות מחמד בבעלותם או שאומצו על ידם, המקלה על ניהול הקשר בין חיות המחמד לבעליהן.

**נתונים מוכלים:**

- תעודת זהות בעלים

- שם

- כתובת

- רשימת חיות מחמד בבעלות או מאומצות

**פונקציונליות:**

- יצירת רשומות בעלים חדשות

- הוספה או הסרה של חיות מחמד מרשימת הבעלים

- הצגת פרטי הבעלים

- ניהול אחסון ואחזור נתוני הבעלים

1. AdoptionRequest

תיאור: מבנה זה חיוני לניהול תהליך האימוץ. הוא מתעד ועוקב אחר כל בקשת אימוץ, מקשר בין חיות מחמד למאמצים פוטנציאלים ושומר היסטוריה של כל פעילויות האימוץ.

**נתונים מוכלים:**

- מזהה בקשה

- פרטי חיית המחמד

- פרטי המאמץ הפוטנציאלי

- סטטוס בקשת האימוץ

- תאריכים של אירועים משמעותיים בתהליך האימוץ

**פונקציונליות:**

- יצירת בקשת אימוץ חדשה

- עדכון ומעקב אחר סטטוס אימוצים בתהליך

- שמירת רשומות מפורטות של פעילויות אימוץ

- אחזור וסקירת בקשות אימוץ בעבר

1. DonationRecord

תיאור: מבנה ה`DonationRecord` מיועד לעקוב אחר תרומות כספיות למקלט. הוא מתעד פרטים של תורמים ותרומות, ומסייע בניהול הכספי ותכנון המקלט.

**נתונים מוכלים:**

- מזהה תרומה

- פרטי התורם

- סכום התרומה

- תאריך התרומה

**פונקציונליות:**

- תיעוד תרומות חדשות

- ניהול מידע על תורמים

- מתן סיכומים לדיווחים פיננסיים

- הבטחת דיוק ברישום ומעקב אחר תרומות כספיות

1. CommandRecord

תיאור: מבנה זה משמש לתיעוד ומעקב אחר פקודות או פעולות אילוף של חיות מחמד במערכת.

**נתונים מוכלים:**

- שם פקודה

- רמת ביצוע הפקודה

- התאריך שבו הפקודה נלמדה

**פונקציונליות:**

- תיעוד פקודות אילוף חדשות

- אחזור היסטוריית פקודות

- הצגת רשומות פקודה

1. Shelter

תיאור: המבנה `Shelter` מייצג את מקלט חיות המחמד עצמו, וכולל מידע על פעילותו, מלאי חיות המחמד והצוות שלו כו'..

**נתונים מוכלים:**

-שם

-כתובת

- רשימת חיות מחמד

- רשימת מתנדבים

- רשומות פיננסיות

-רשימת מתנדבים

פונקציונליות:

- ניהול מלאי חיות מחמד

- תיאום פעילויות מתנדבים

- טיפול בעסקאות פיננסיות

1. Volunteer

תיאור: מבנה זה מפרט מידע על מתנדבים במקלט, כולל המידע האישי שלהם ותפקידיהם בארגון.

**נתונים מוכלים:**

- מזהה מתנדב

- שם

- תפקיד במקלט

- שעות התנדבות

**פונקציונליות:**

- רישום מתנדבים חדשים

- הקצאת תפקידים ואחריות

- מעקב אחר שעות מתנדבים

- ניהול פרופילים של מתנדבים

1. PetMedicalRecord

תיאור: מבנה ה-PetMedicalRecord מכיל את כל המידע הרפואי הרלוונטי עבור חיית מחמד ספציפית. הוא כולל פרטים על חיסונים, מצבים רפואיים וביקורים אצל הווטרינר. המבנה מאפשר גישה קלה ועדכון של נתונים רפואיים, ומספק תמונה מקיפה על בריאותה של חיית המחמד.

**הערה חשובה:**

**מבנה זה הוא מבנה "אקסטרה", במבנה זה לא הספקתי ליישם בצורה מיטבית את פונקציונליות הטעינה מקבצים על אף שהקוד קיים ועובד.**

**מנגד במבנה זה לקחתי על עצמי נושא ללימוד עצמי והוא כתיבת קבצים לתיקיות ופתיחת תיקיות עבור קבצים. עבור כל חיית מחמד שלה קיים רישום רפואי נוצרת תיקייה ייעודית המכילה את מספר זהות של חיה המחמד וכך למעשה החזון היה להשתמש במבנה נתונים שונה מהמבנים הסטנדרטים כמו מערך או רשימה מקושרת בשביל לשמור את כלל הרשומות הרפואיות.**

**נתונים מוכלים:**

מצביע לאובייקט Pet המשויך

מחרוזת המכילה את החיסונים של חיית המחמד

מחרוזת המכילה את המצב הרפואיים של חיית המחמד

מצביע לאובייקט Date המייצג את הביקור האחרון של חיית המחמד אצל הווטרינר

**פונקציונליות:**

יצירת רשומה רפואית חדשה לחיית מחמד עם הפרטים הנתונים

עדכון רשומה רפואית קיימת עם פרטים מעודכנים על חיסונים, מצבים רפואיים וביקורים אצל הווטרינר

הצגת הפרטים של רשומה רפואית של חיית מחמד

שחרור הזיכרון שהוקצה עבור רשומה רפואית של חיית מחמד

יצירת רשומה רפואית חדשה לחיית מחמד מקלט משתמש

שמירה וטעינה של רשומות רפואיות של חיות מחמד מקבצי טקסט ובינארי

1. DataTrainingTips

תיאור: מבנה זה מספק נתונים מובנים עבור טיפים והנחיות אימון, המשמשים לחינוך בעלי חיות מחמד ומתנדבים לגבי טיפול נכון בחיות מחמד.

**נתונים מוכלים:**

- גזע חיית המחמד

- קטגוריית גיל

- טיפ

**פונקציונליות:**

- מתן תוכן חינוכי מותאם אישית עבור בעלי חיות המחמד

- התאמה אישית של טיפים לקהלים וחיות מחמד שונים

**ניתוח יעילות וטכניקות דחיסה שיושמו במבנה CommandRecord:**

במהלך פיתוח הקוד לאחסון נתוני CommandRecord בקבצים בינאריים, השתמשתי במספר טכניקות דחיסה כדי למטב את השימוש בשטח האחסון.

1. דחיסת שם הפקודה:

- החלטתי להשתמש ב-uint8\_t במקום סוג מספר שלם גדול יותר כדי לאחסן את אורך שם הפקודה. זה מגביל את האורך המקסימלי ל-255 תווים, אך זה מספיק ברוב המקרים.

- הגישה הזו חוסכת לנו מקום בקובץ הבינארי בהשוואה לשימוש בסוגי מספרים שלמים גדולים יותר.

2. דחיסת תאריך למידה:

- כדי לחסוך מקום, החלטתי לקודד את היום והחודש של תאריך הלמידה לתוך uint8\_t יחיד.

- השתמשתי ב-5 הסיביות התחתונות (0-4) לאחסון היום (ערכים 0-31) וב-3 הסיביות העליונות (5-7) לאחסון החודש (ערכים 0-7).

- הטכניקה הזו מאפשרת לנו לדחוס את תאריך הלמידה מכמה משתני int נפרדים (ליום ולחודש) ל-uint8\_t יחיד.

- השתמשתי בפעולות סיביות כמו הזזה ו-AND כדי לקודד ולפענח את התאריך בצורה יעילה.

3. דחיסת שנת למידה:

- כדי לצמצם את גודל האחסון של שנת הלמידה, החלטתי לחסר שנת בסיס (2000) מהשנה לפני שמירתה בקובץ הבינארי.

- הגישה הזו מאפשרת לנו לייצג את השנה כערך יחסי קטן יותר, מה שחוסך לנו סיביות.

- בזמן טעינת הנתונים, אנחנו פשוט מוסיפים את שנת הבסיס בחזרה כדי לקבל את השנה המקורית.

אחוז דחיסה משוער:

בהתבסס על כמה הנחות, כמו אורך ממוצע של 20 תווים לשם פקודה, הערכתי את אחוז הדחיסה שמושג על ידי הטכניקות האלה:

- שימוש ב-uint8\_t במקום int לאורך שם הפקודה חוסך 3 בתים לרשומה.

- דחיסת תאריך הלמידה ל-uint8\_t יחיד חוסכת בית אחד לרשומה.

- דחיסת שנת הלמידה על ידי חיסור שנת הבסיס חוסכת 2 בתים לרשומה.

סך הכל, אנחנו חוסכים בערך 6 בתים לכל רשומה. בתרחיש שבו CommandRecord היה דורש בדרך כלל 32 בתים ללא דחיסה, הגרסה הדחוסה תדרוש רק 26 בתים. לכן, אחוז הדחיסה המשוער הוא:

אחוז דחיסה = (גודל לא דחוס - גודל דחוס) / גודל לא דחוס \* 100

= (32 - 26) / 32 \* 100

= 18.75%

חשוב לציין שזהו אומדן בלבד, והתוצאות בפועל עשויות להשתנות בהתאם לנתונים הספציפיים.

לסיכום, הטכניקות שיישמתי בקוד כוללות:

1. שימוש ב-uint8\_t לאחסון אורך שם הפקודה.

2. קידוד היום והחודש של תאריך הלמידה לתוך uint8\_t יחיד באמצעות פעולות סיביות.

3. דחיסת שנת הלמידה על ידי חיסור שנת בסיס.

הטכניקות האלה מביאות להפחתה משמעותית בגודל האחסון הדרוש לכל CommandRecord, עם אחוז דחיסה משוער של כ-18.75%. עם זאת, חשוב לזכור שיש לאזן בין דחיסה לבין קריאות ותחזוקה של הקוד. השימוש בפעולות סיביות ובערכים מקודדים עלול להקשות על הבנת הקוד בעתיד אך זוהי דרישת הפרויקט.

אני מקווה שהניתוח הזה מספק הסבר ברור על הטכניקות שיישמתי כדי למטב את אחסון הנתונים בפרויקט שלי.

שרטוט הפרויקט:

מצורף כקובץ