****

**המחלקה למדעי המחשב**

שם הפרויקט: "FITNESSSMART"- פיתוח אפליקציית כושר

Project Name: "SMART FITNESS" - Mobile fitness application

PDD

|  |  |
| --- | --- |
| שמות הסטודנטים: | עומרי אמסלם, ולדיסלב פופוב |
| מספר תעודת זהות: | 318883386, 327391413 |

**הקדמה**

כמו שציינו במספר SOW, אנחנו הולכים לפתח אפליקציית כושר חדשה. האפליקציה תאפשר למשתמשים לעקוב אחר אימוני הכושר שלהם, לקבל המלצות לתוכניות אימונים מותאמות אישית

**ארכיטקטורה**:

Client-Server עם לקוח (Client App) ושרת (Server)

**רכיבי לקוח (Client):**

1. User Interface - ממשק משתמש להצגת התפריטים, שאלונים, רשימות תרגילים וכו'.

2. User Authentication - מודול לניהול משתמשים רשומים/אורחים.

3. Network Layer\* - לתקשורת עם השרת.

**רכיבי תשתית (infrastructure):**

1. API Layer\* - לקבלת בקשות מהלקוח ושליחת נתונים.

2. User Management - ניהול משתמשים רשומים והעדפותיהם.

3. Exercise Logic - הלוגיקה העיקרית של האפליקציה:

- קבלת נתוני המשתמש והעדפותיו

- מנגנון המלצות לתרגילים על סמך הנתונים

- כללים לאזהרות (תרגילים חוזרים על אותו שריר וכו')

4. Exercise Database - מאגר התרגילים והמידע הרלוונטי (שרירים, קטגוריות וכו').

5. Users Database- מאגר משתמשי האפליקציה ופרטי התחברות.

6. Authentication Layer - לאימות משתמשים רשומים.

ארכיטקטורה זו תאפשר ניהול נכון של המשתמשים, הפרדה ברורה בין הלקוח והשרת, ומימוש הלוגיקה העיקרית של האפליקציה בצד השרת באופן יעיל ומאובטח.

**\*מימוש האפליקציה שלנו לא יכלול צד שרת ו API אלא בסיס נתונים לוקאלי המתממשק לאפליקציה**

**אלגוריתם עבור האפליקציה:**

1. **הרשמה וניהול משתמש**:

- המשתמש יכול להיכנס כאורח או ליצור חשבון משתמש חדש.

- לאחר ההרשמה, המשתמש ימלא שאלון לאיסוף פרטים אישיים, מטרות כושר, העדפות ומגבלות.

- הנתונים האישיים והעדפות ישמרו בפרופיל המשתמש.

2. **בניית תוכנית אימונים אישית:**

- המשתמש יבחר את התחומים המועדפים עליו.

- המערכת תציע למשתמש תחומי אימון שונים (כוח, אירובי, חיטוב וכו') על בסיס העדפותיו.

- המערכת תציג רשימה של תרגילים מתאימים לתחומים הנבחרים.

- המשתמש יוכל לבחור את התרגילים הרצויים ולהוסיפם לתוכנית האימונים האישית שלו.

- המשתמש יוכל להתאים את מספר החזרות, המשקלים וכו' לכל תרגיל על פי רצונו.

3. **התאמה אישית ומעקב:**

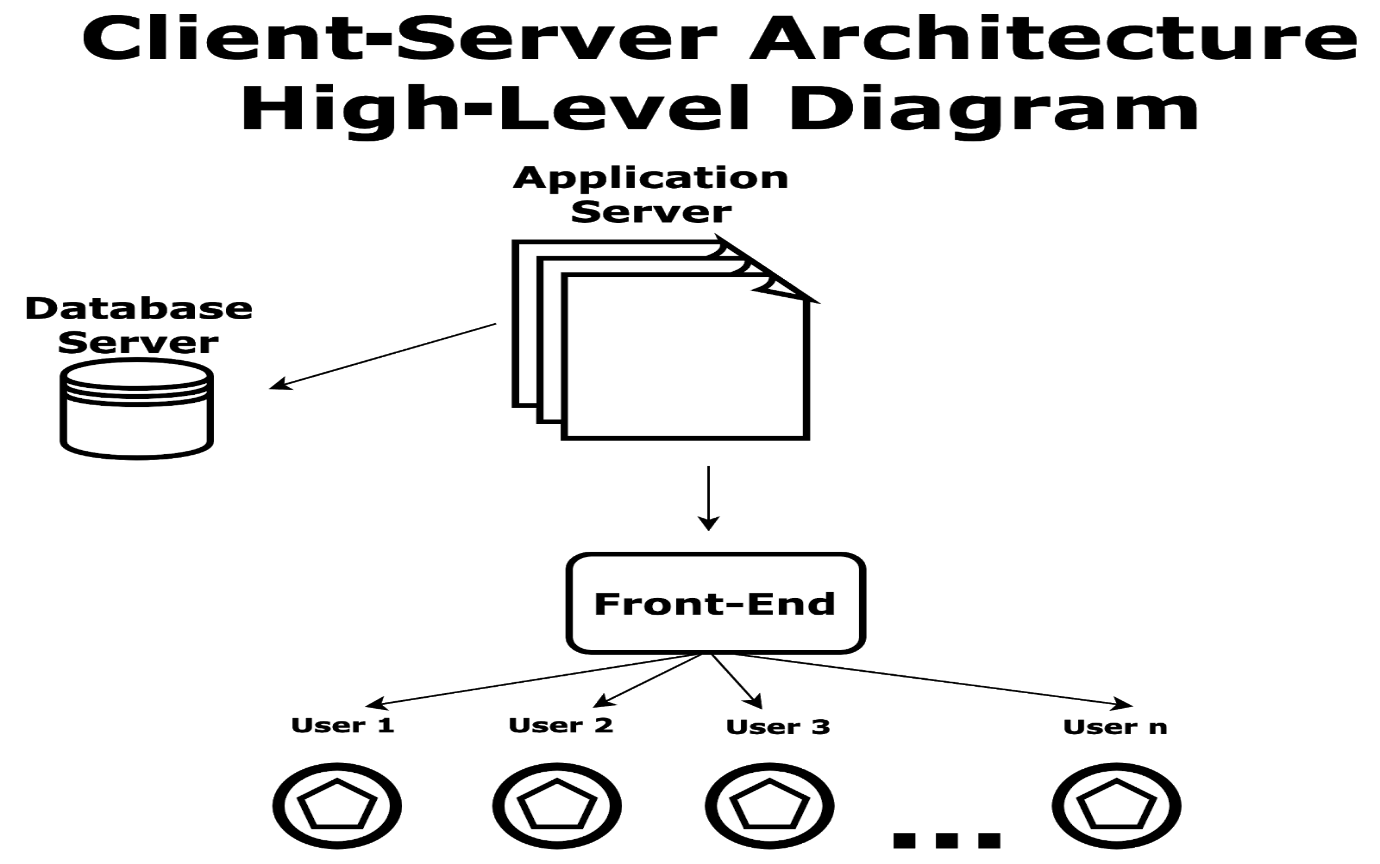
- המערכת תנטר את התקדמות המשתמש ותציע התאמות לתוכנית האימונים על בסיס ביצועיו.

- אם המשתמש מתקשה בתרגיל מסוים, המערכת תציע תרגילי חלופה או התאמות.

- אם המשתמש מתמקד יתר על המידה באזורי גוף מסוימים, המערכת תציע תרגילים לאיזון.

- המשתמש יוכל לערוך את תוכנית האימונים שלו בכל עת על בסיס ההתקדמות וההעדפות המשתנות שלו.

**תיאור גרפי של הארכיטקטורה**



**תהליך העבודה:**

1. המשתמש ייכנס לממשק הלקוח כאורח או משתמש רשום.

2. במקרה של אורח, יוצג שאלון להזנת נתונים והעדפות.

3. האפליקציה תעבד את הבקשה דרך ה-Exercise Logic בשילוב עם ה-Exercise Database.

4. המשתמש יכניס את התרגילים שהוא מבצע בדרך כלל.

5. השרת יחזיר רשימת תרגילים מומלצים.

5. האפליקציה תציג את התרגילים בממשק.

6. המשתמש יבחר תרגיל, והבחירה תיבדק על ידי האפליקציה.

7. תוחזר הודעות רלוונטיות אם יש.

**- Exercise Logicאלגוריתם חישובי למעקב אחרי מספר התרגילים שמשתמש עושה על אותו שריר על מנת ולוודא שהוא תואם למטרות האימון שלו.**

1. **מיפוי תרגילים לשרירים:**

- ליצור מסד נתונים או טבלה שמקשרת כל תרגיל לשרירים העיקריים שהוא מפעיל.

- למשל, לחיצת ארנולד יקושר לשרירי הכתפיים, החזה והזרועות.

2. **ניהול תוכנית האימונים האישית**

- בעת בניית התוכנית האישית, ישמר מונה לכל שריר העוקב אחר מספר התרגילים שנבחרו לשריר זה.

- כאשר משתמש מוסיף תרגיל לתוכנית, יעלו המונים של השרירים הרלוונטיים.

3. **בדיקת איזון התוכנית:**

- יוגדר טווח מומלץ לכמות התרגילים לכל שריר, בהתבסס על מטרות האימון של המשתמש.

לדוגמה, אם המטרה היא בניית מסה שרירית, טווח של 9-12 חזרות לשריר ראשי יכול להיות מומלץ.

- יבדק אם המונים של השרירים נמצאים בטווח המומלץ.

4. **התראות והמלצות:**

- אם מונה של שריר חורג מהטווח המומלץ (יותר מדי או מעט מדי), תוצג התראה למשתמש.

- ההתראה יכולה להציע המלצות להוספה/הסרה של תרגילים לאיזון התוכנית.

- ניתן גם להציע תרגילי חלופה לשרירים מסוימים על מנת ליצור איזון טוב יותר.

5. **עדכון שוטף של המונים:**

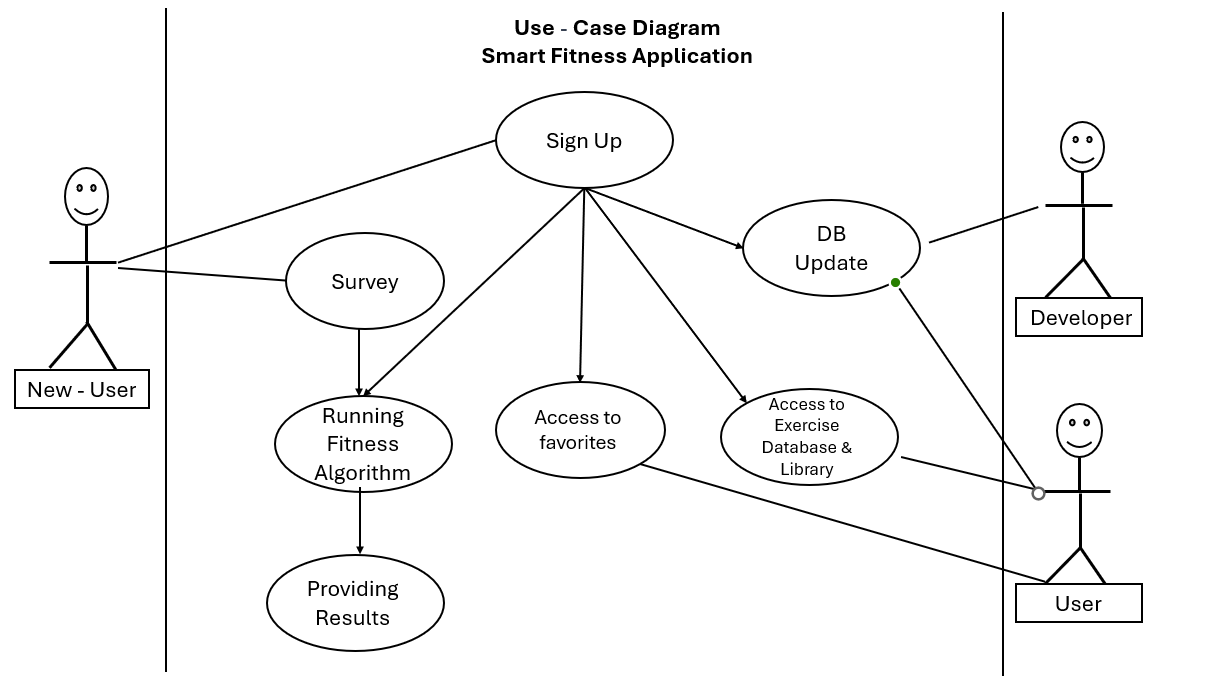
- לאחר כל אימון, יאופשר למשתמש לסמן את התרגילים שביצע.

- המונים יעודכנו בהתאם לכמות התרגילים שנעשתה.

- זה יאפשר מעקב שוטף על ההתקדמות ביחס לתוכנית האימונים המקורית.

באמצעות אלגוריתם זה, האפליקציה תוכל לוודא שתוכניות האימונים האישיות שמשתמשים בונים הן מאוזנות ותואמות למטרות הכושר שלהם. זה יסייע למנוע עומס יתר על שרירים מסוימים ויבטיח התקדמות מושכלת לקראת היעדים המבוקשים.

**תיאור גרפי UML של מקרי שימוש:**



|  |  |  |  |
| --- | --- | --- | --- |
|  | **שם ה USE CASE** | **תיאור** | **שחקן ראשי** |
| 1 | התחברות למערכת | בתהליך זה המשתמש יזין פרטים מזהים על מנת להתחבר למערכת. המערכת תבדיל בין 2 סוגי משתמשים: משתמש חדש, משתמש קיים. על פי סוג המשתמש יתאפשרו אפשרויות שונות במערכת. | משתמש קיים, משתמש חדש |
| 2 | עדכון DB | אפשרות זו תהיה זמינה עבור משתמש. בתהליך זה משתמש יוכל להוסיף ולעדכן תרגילים במאגר | משתמש קיים |
| 3 | הצגת מאגר תרגילים הכולל | אפשרות זו תהיה פתוחה למשתמשים. יוצג כל מאגר שך התרגילים ללא הפעלת אלגוריתם התאמה. | משתמש קיים |
| 4 | מענה על שאלון ראשוני | אפשרות זאת תהיה קיימת למשתמשים חדשים בלבד. המערכת תציג בצורה אוטומטית לאחר התחברות ראשונית שאלונים עליהם יענה המשתמש | משתמש חדש |
| 5 | הפעלת אלגוריתם התאמה | לאחר שהמשתמש הזין את כל הפרטים הנדרשים בשאלונים הראשוניים, המערכת תפעיל אלגוריתם אשר יעריך את אחוזי ההתאמה של התרגילים למטרת האימון  בנוסף, תהליך זו יופעל כאשר משתמש קיים יבחר להפעיל אותה. | משתמש חדש, משתמש קיים |
| 6 | הצגת תוצאות | הצגת תוצאות הפעלת אלגוריתם ההתאמה על המסך, כאשר התוצאות יסודרו על פי מידת ההתאמה. המשתמש יוכל להיכנס לתוצאות ולראות פרטים נוספים התריגלים. המשתמש יוכל לשמור תוצאות כרצונו לרשימת מועדפים ולחזור אליה בכל עת. | משתמש קיים, משתמש חדש |
| 7 | הצגת מועדפים | המשתמש לאחר התחברות יוכל לחזור אל רשימת התרגילים אשר סימן כמועדפים עליו. | משתמש קיים. |

**Activity Diagram- התחברות משתמש למערכת:**

Check if new user

Present survey

Activate algorithm.

Present algorithm results

Save Exercise to favorites.

Show message “Exercise already saved”

[False]

[False]

[False]

[False]

Show the user’s favorites

[False]

**תרשים עץ המסכים**

מסך התחברות ראשי

שאלון משתמש חדש

מאגר כללי של התרגילים

מועדפי משתמש

מסך תוצאות ההתאמה

פרטי תרגיל

מסך עריכה ראשי

מסך עריכת תרגיל

מסך הוספת תרגיל חדש

**תרשים GANT**

