

מסמך דרישות

מערכת ניהול משחקים וחלוקת שחקנים לקבוצות – מסמך דרישות ראשוני

מטרה

המערכת נועדה לאפשר יצירה וניהול של משחקי ספורט קבוצתיים חובבניים, כולל הזנת פרטי שחקנים, חלוקתם לקבוצות מאוזנות לפי דירוגים ותפקידים, וניהול המשחקים בענן – ללא צורך בשמירת מידע מקומית. המערכת מבוססת על תשתית Serverless של AWS ומיועדת לשימוש קל ונגיש מכל מכשיר.

דרישות פונקציונליות

1. יצירת משחק חדש

- טופס הכולל: תאריך ושעה.
- מזהה ייחודי `game_id` נוצר אוטומטית.
- שמירת המשחק בטבלת `games` (DynamoDB) עם `team_id` ו-`created_by` (כרשימה פתוחה)

2. הזנת פרטי שחקנים

- המשתמש מזין את מספר השחקנים (בין 4 ל-33).
- בהתאם, מופיעות רק אפשרויות לחלוקה שווה (למשל: 16 שחקנים $\rightarrow 2, 4, 8$ קבוצות).
- עבור כל שחקן נדרש להזין:
 - שם (חייב להיות ייחודי).
 - עמדה (שוער, בלם, מגן, קשר, חלוץ).
 - דירוג מ-1 עד 10 בקפיצות של 0.5.
- טבלת שחקנים מוצגת עם אפשרות עריכה `inline`.
- הנתונים נשמרים ב- `DynamoDB (players_new)` רק לאחר לחיצה על "שמור".

3. חלוקת שחקנים לקבוצות

- לאחר שמירת כל השחקנים, כפתור "חלק לקבוצות" מופעל.
- מופעלת פונקציית `Lambda` שמשתמשת ב- `GPT-4o` לצורך חלוקה חכמה לפי דירוגים ותפקידים.
- כל קבוצה מקבלת שוער אחד בלבד.

○ תוצאות החלוקה נשמרות:

- `team_id` לכל שחקן בטבלת `players_new`.
- רשומות חדשות בטבלת `teams` עם ממוצע דירוג.

4. ניהול אבטחה וגישה

- כל משתמש מזוהה עם Amazon Cognito.
- המידע נשמר תחת המשתמש שיצר את המשחק.
- ההרשאות מוגבלות בהתאם ל- ID Token.

5. ממשק משתמש

- מבוסס HTML + JS, מאוחסן ב- S3.
- כולל עמודי `index.html` ו- `login.html`.
- אינטראקציה עם המערכת נעשית באמצעות API Gateway ו- Lambda בלבד.

מבנה בסיסי של טבלאות DynamoDB

- `games`: `game_id`, `datetime`, `created_by`, `players_count`, `teams_count`, `team_id` (array)
- `players_new`: `player_id`, `game_id`, `name`, `position`, `rating`, `team_id`
- `teams`: `team_id`, `game_id`, `average_rating`

אבטחה

- IAM Roles מגדירים הרשאות בין Lambda לשירותי AWS.
- משתמשים מזוהים באמצעות Cognito עם JWT Token.
- אין גישה למידע ללא Token תקף