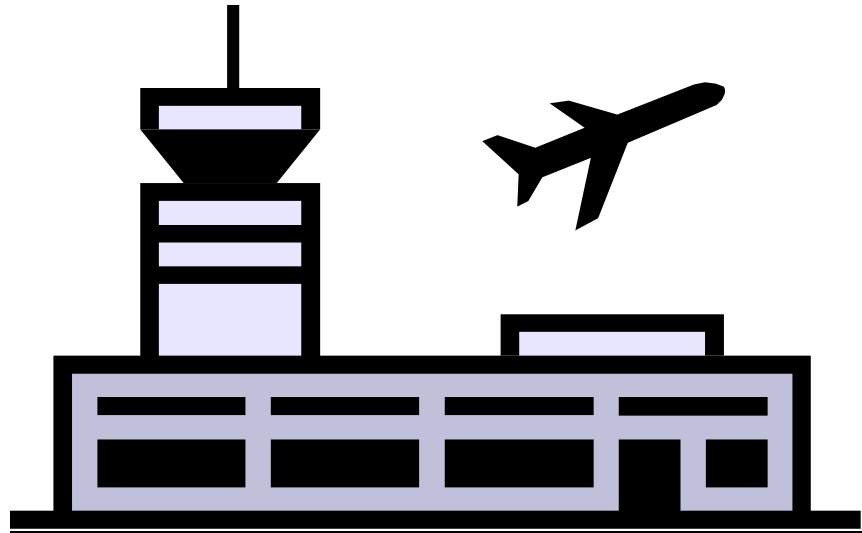


פרויקט סיום



שדה תעופה

פרויקט סיום מסלול תוכנה – מערכת מדמה מגדל פיקוח בזמן אמת

דרישות המערכת

אנו בונים מערכת מחשוב שתדמה פעילות בשדה תעופה .

המערכת תכלול את הרכיבים:

- שכבה לוגית (שתנהל את שדה התעופה)
- Services (שיאפשרו גישה מרוחקת לשכבה הלוגית)
- User Interface (UI) (שיציג את מצב השדה באופן ויזואלי)
- סימולטור (יצור המראות ונחיתות וישלח אותם לשכבה הלוגית דרך ה Service)
- Tests (בדיקות לשכבה הלוגית – יחליף את הסימולטור עם נתונים ידועים ויבדוק מצבים לוגים אפשריים)

בשכבה הלוגית יש 3 רכיבים מרכזיים:

- מטוס
- תחנה במסלול
- תמונת מצב כוללת של כל התחנות + המטוסים שנמצאים בכל תחנה

DTOs (Data Transfer Objects)

תמונת המצב הכוללת של השכבה הלוגית צריכה להישלח (בכל פעם שיש שינוי) ל UI.

לכן האובייקטים שישלחו צריכים להיות מאוד פשוטים (להכיל Data בלבד ולא לוגיקה), וכמו כן הם צריכים לעבור Serialization כאשר הם נשלחים (הSerialization מטופל ברמת ה Service באופן אוטומטי, אך ה DTOs צריכים לתמוך בכך [לדוגמה להשתמש ב Attributes מתאימים]).

בניית השכבה הלוגית

Testability

השכבה הלוגית צריכה להבנות באופן שיאפשר Unit Test כלומר ליצור Interface עבור כל התלויות (לדוגמה גישה ל DB, Service, Simulators וכדומה).

השכבה הלוגית צריכה לעבוד מול ה Interface ולא מול המימוש, כך שבאמצעות Test יהיה ניתן לבדוק את הלוגיקה.

יש לשים לב שכל תת לוגיקה שדוגמת זמן צריכה גם היא להיות מופרדת ל interface כדי שה Test לא יהיה תלוי בזמן של שלא בשליטתו.

מאפייני תתי היחידות הלוגיות

מטוס

מרגע שמטוס נכנס לתחנה הוא צריך להמתין זמן (שרירותי) ולידע את התחנה שהוא מוכן למעבר לתחנה הבאה.

התחנה

הלוגיקה של כל התחנות זהה ולכן הגיוני יותר לטפל בלוגיקה של תחנה בודדת ולא של כל שדה התעופה ביחד.

התנהלות שדה התעופה הי למעשה כלל הפעילות שמתבצעת בכל התחנות.

הגדרת הלוגיקה:

- תחנה יכולה להיות מחוברת לאחת או יותר תחנות המשך (החיבורים האלה יוצרים את המסלול).
- לא יכולים להימצא שני מטוסים בו זמנית בתחנה
- תחנה מקבלת מטוס וממתינה עד שהמטוס מוכן לעבור לתחנה הבאה.
- בשלב זה התחנה בודקת אם אחת מתחנות ההמשך פנויה.
 - במידה שאחת מתחנות ההמשך פנויה היא מעבירה אליה את המטוס והתהליך מתחיל מחדש.
 - במידה שאף אחת מהתחנות ההמשך אינה פנויה היא מאזינה עד שאחת התחנות מתפנה ומעבירה עליה את המטוס.
 - במידה וזו תחנה אחרונה היא פשוט שוכחת את המטוס

תמונת המצב של שדה התעופה

מייצגת כל שינוי בסטטוס של התחנות.

כאשר יש שינוי הוא משודר ל UI (דרך Service) ונשמר ל DB (דרך Repository).

כאשר האפליקציה עולה היא טוענת את המצב העדכני דרך ה Repository.

רשימת המראות ונחיתות

מעבר לשדה עצמו קיימות שתי רשומות:

- המראות מתוכננות
- נחיתות מתוכננות

שתי הרשימות הנ"ל מחזיקות מידע על מספר הטיסה.

המטוסים נכנסים לרשימות ע"י Service וכל שינוי ברשימה נשמר ל Repository ומשודר ל UI.

בעליית התוכנה הרשימות מתעדכנות מה Repository עם נתונים שעדיין לא עבר זמנם.



שירותים נוספים

ה Logic יפתח אפשרות לתשאול של המצב הכולל ברגע נתון שיכלול את מצב התחנות והרשימות (ה UI ישתמש בשרות הזה דרך Service כאשר הוא עולה).

סימולטור

לוגיקה שתייצר המראות ונחיתות, תעבוד מול ה Service שחושף את ה Logic.

Database

יהיה חשוף דרך Repository ויחזיק את המצב העדכני של שדה התעופה.
צריך להכיל 4 טבלאות:

- זמן אמת של מצב התחנות (שורה עבור כל תחנה ועדכונים של מצב התחנה)
- היסטוריה: כל שינוי של מטוס בתחנה את הזמן שהוא נכנס לתחנה והזמן שהוא יצא מהתחנה
- המראות מתוכננות (לא למחוק נתונים, ה query צריך להביא נתונים עתידיים בלבד)
- נחיתות מתוכננות (לא למחוק נתונים, ה query צריך להביא נתונים עתידיים בלבד)

Services

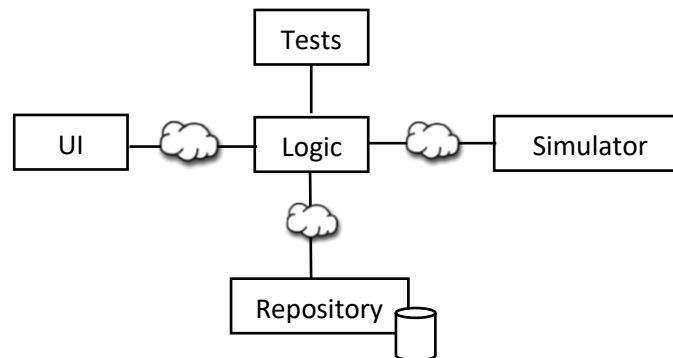
כל הממשקים בין היחידות השונות חשופים דרך Services.

UI

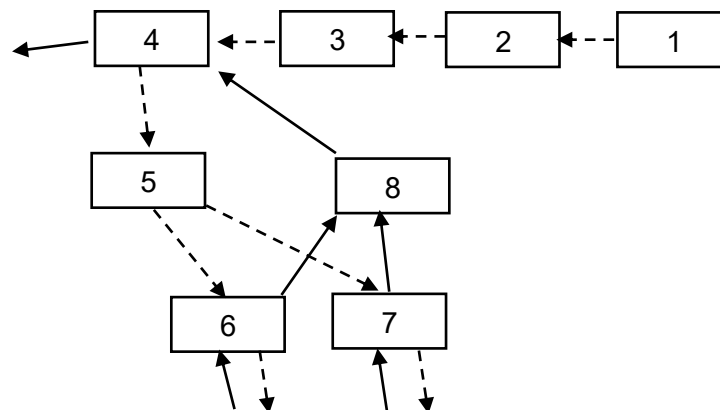
מציג את

- רשימות ההמראות והנחיתות
- מצב עדכני של השדה בתצוגה טבלאית
- תצוגה גרפית של השדה (בשלב שני של הפיתוח אחרי שכל השאר עובד)

תרשים התחנות



מסלולים



← המראה

← - - נחיתה

Definition of Done

שלב א' (Unit Tests)

שלב א' מסתיים כאשר הלוגיקות הבאות נבדקו ע"י Unit Tests בשכבה הלוגית (לפני מימוש Simulator, Repository or UI):

- טיסה לא מתקבלת כאשר תחנה אינה פנויה.
- טיסה מתקבלת כאשר תחנה פנויה.
- טיסה לא עובר בין תחנות כאשר אין תחנות המשך פנויות.
- תחנה הופכת לריקה לאחר שמטוס ששהה בה מוכן לעבור תחנה ויש תחנת המשך פנויה.
- כאשר לתחנה לא משויכת תחנת המשך היא מתפנה כאשר מטוס ששהה בה סיים את זמן שהיה.
- כאשר תחנה מעבירה טיסה לתחנת המשך שאינה פנויה, אך תתפנה בעתיד, המטוס יעבור לתחנה רק בזמן שהיא מתפנה (לא לפני).
- כאשר מחוברות לתחנה, תחנות המשך פנויות של המראות וגם של נחיתות, מטוס ממריא יגיע לתחנות של המראות ומטוס נוחת יגיע לתחנה של נחיתות.
- כאשר תחנה מחוברת לתחנות המשך של המראה שאינן פנויות ולתחנות המשך של נחיתה פנויות, המראה לא תעבור לשום תחנת המשך.
- הוספה של טיסה נכנסת נכתבת לממשקי ה Repository and UI (לאחר אישור של תחנת הכניסה).