Final Project

**Flight Control**

****

**הקדמה**

**מטרה ומבט כללי**

מסמך זה מתאר דרישות המערכת ל"flight control" ,

"flight control" הינו web-service המסמלץ פעילות בשדה תעופה ומאפשר ליוזרים לראות את המצב הנוכחי של שדה התעופה מבחינת – המראות, נחיתות, הסעות, חניות מטוסים, והסטוריית נחיתות\המראות.

**דרישות פונקציונליות**

"flight control" היא אפלקציה המסמלצת התנהלות שדה תעופה אמיתי ומאפשרת למשתמשים בה לראות באופן איורי את המצב הנוכחי של שדה התעופה באופן מפורט ומלא.

**שכבת התצוגה**

שכבת התצוגה תיהיה מורכבת מתפריט נווט וחלון משתנה בהתאם.

**תפריט הנווט**

תפריט הנווט יהיה מורכב מכפתורים המאפשרים להחליף בין מסכים – "flight control" ו -"flight history"

גרסא עתידית: כפתור "login, logout" (האפלקציה תכלול מערכת התחברות).

**מסך "flight control"**

מסך זה ייצג באופן איורי את המצב הנוכחי של שדה התעופה, תבנית מראה המסך תראה כך:

מסלול

1

2

3

4

5

6

7

8

9

טרמינל

גישה לנחיתה

אחרי ההמראה

מסלול הסעה

מסלול הסעה

מסלול הסעה

-המספרים מייצגים את מספר התחנה

גישה לנחיתה - מנקודה 1 לנקודה 2 ו 3 לפי הסדר.

נחיתה - לנקודה 4

הסעה לעמדת חניה - ל 5 (הסעה) ולפינוי בעמדה 6 או 7. אם עמדה תפוסה המטוס ימתין בנקודה.

המראה:

מטוס שאמור להמריא יתחיל (יופיע) בנקודה 6 או 7 ואז יסיע לנקודה 8 משם ל 4 להמראה ואח"כ ל 9 (אחרי המראה).

בעת מצב שבו קיים מטוס בתחנה מסויימת באותה תחנה תופיע תמונה של אותו מטוס.

**מסך "history flight"**

במסך זה יופיעו טבלאות המראות את כל הסטוריית הטיסה של שדה התעופה.

\*טבלאת המראות – תכיל את כל ההמראות כולל מספר מטוס , שעה ותאריך.

\*טבלאת נחיתות - תכיל את כל הנחיתות כולל מספר מטוס, שעה ותאריך.

גרסא עתידית:

\*בעת לחיצה על מטוס ב"flight control" ייפתח מסך המכיל את כל פרטי המטוס

(שם חברת התעופה, מס' מטוס, מס' נוסעים,יעד ומקור, כמות דלק)

\*ב"flight control" יסומנו מקרים חריגים בהתאם (נחיתת אונס, מחסום במסלול הסעה וכו')

**שכבת האפליקציה**

**בקשות ותגובות Client**

**בעת פתיחת האפלקציה הclient שולח בקשת login לסרבר ומקבל response בהתאם.**

**אם הlogin הצליח מעתה והלאה הclient יצטרך לטפל בcallbacks המגיעים מהסרבר**

**בקשת login\logout**

כשהclient שולח בקשת login הוא מחכה לresponse מצד הסרבר שהlogin הצליח ויקבל את כל הסטוריית המראות ונחיתות של שדה התעופה כולל המצב הנוכחי של שדה התעופה:

* שדה התעופה יכיל את הפרטים הבאים
  + מסלולים
    - מסלול המראה/נחיתה
    - מסלול לפני חניה
    - מסלול לפני נחיתה
    - מסלול לפני המראה
    - בכל מסלול יכול להיות מטוס נוסע אחד בכל רגע
    - יהיה אפשר לשנות את מבנה המסלול (מראש) והאפליקציה תדע להציג ולנהל את שדה התעופה
  + נקודת המתנה
    - באוויר בין מסלולי לפני נחיתה יש נקודות המתנה בהם מטוסים יכולים להסתובב במקום לגובה לפני שמקבלים אישור לנחות
  + מטוסים
    - מטוס נוסעים
      * מעלה נוסעים לפני המראה
      * מוריד נוסעים לפני חניה
    - מטוס מטען
      * טעינה לפני המראה
      * פריקה לפני חניה
    - מהירות טיסה
    - כמות דלק
  + טיסות
    - פרטי הטיסה
    - מסלול הטיסה
    - תהליך הטיסה
  + תהליך
    - יכיל באיזה מסלולים ובאיזה סדר על המטוס לעבור בטיסה
  + חניות
    - חניה מוגבלת בכמות מטוסים
  + מקרי חרום
    - נחיתת אונס
      * ברגע שמטוס מודיע למגדל הפיקוח על נחיתת אונס מגדל הפיקוח ידאג לפנות את המסלולים ולקדם את המטוס לנחיתה
    - שריפה במסלול המראה ונחיתה
      * אחת לכמה המראות/נחיתות יכולה לפרוץ שריפה שתגרום למסלול להיות תפוס ליותר זמן עד לטיפול בשריפה וחזרה למצב שגרה
    - יכולים להתרחש ידנית ע"י המשתמש או אוטומטית ע"י הסימולטור

-הclient ימיר את כל הפרטים של המצב הנוכחי של שדה התעופה למראה איורי במסך "flight control" כך שהפרטים יסומלצו לתמונה אחת.

-הclient ימיר את כל הפרטים לגבי הסטוריית שדה התעופה למראה טבלאתי מסודר הכולל טבלת המראות וטבלת נחיתות.

בעת יציאה מהאפלקציה בכפתור איקס הclient ישלח בקשת logout לסרבר כדי למנוע callbacks מהסרבר לclient המתנתק

**הודעות callback מהסרבר**

הclient יקבל הודעה מהסרבר ללא שליחת request , זה קורה בכל שינוי שקיים במצב הנוכחי של שדה התעופה

-הclient יקבל את מס' המטוס והשינוי שביצע וירנדר את השינוי לאיור מתאים

-הclient יקבל התראות לגבי נחיתות\המראות שמתרחשות וירנדר את המידע לטבלה המתאימה ב"flight history"

**דרישות סרבר**

* הסרבר צריך לקבל בקשות מclients ולשלוח response בהתאם
* הסרבר "דוחף" הודעות לclients כאשר מתרחש שינוי במצב הנוכחי של שדה התעופה,

ההודעות יכילו את השינוי שהתבצע.

* הסרבר יעדכן את מסד הנתונים בכל שינוי שמתרחש בשדה התעופה

**Login\logout responses**

בעת בקשת login הסרבר ימקם את הclient בlist של callbacks על מנת שיוכל לשלוח לו הודעה כל פעם שמתרחש שינוי וישלח response המכיל את כל הסטוריית שדה התעופה כולל מצב נוכחי.

**Flights Data**

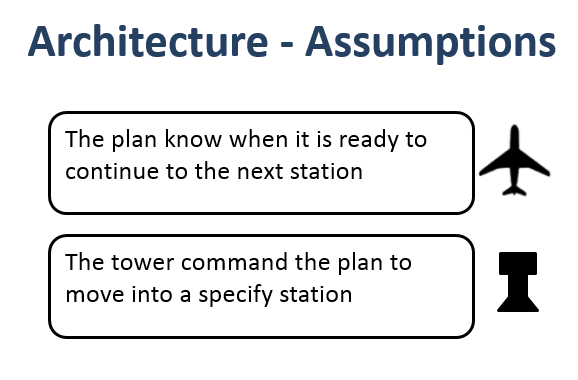
סרבר האפלקציה flight control יכיל גישה לDB בו יישמר כל המידע לגבי המראות, נחיתות ומצב נוכחי של שדה התעופה:

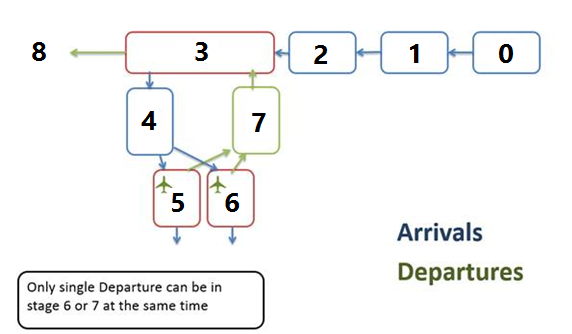
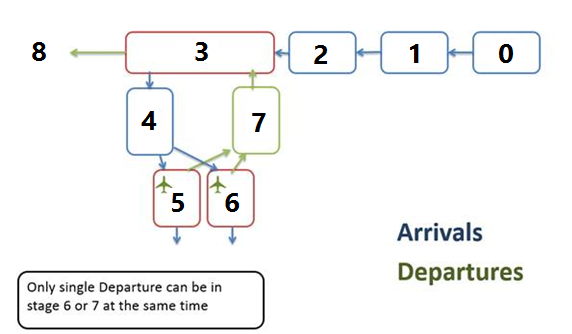
|  |  |
| --- | --- |
| **arrivals** | |
| planId | int |
| Time | DateTime |

|  |  |
| --- | --- |
| **departures** | |
| planId | int |
| Time | DateTime |

|  |  |
| --- | --- |
| **currentState** | |
| planId | int |
| StationId | int |

**ארכיטקטורה**



****

**מבנה המערכת**

Data – מחולק לשניים – 1. סימולטורים שממציאים המראות ונחיתות (לוקחים ממערך לפי זמן) 2. מערכת שמוציאה נתונים מ DB

Logic – מערכת הפונקציונליות של השדה – כוללת מכניזם של תזמון ה"מסלול" ומערך הטיסות באוויר ועל הקרקע.

Entities – ישויות – הגדרתם בלבד

Server – יחזיק את ה State של המערכת וכן את ה Host של ה WCF

WCF – ממשק Service ל GUI

GUI – בהתחלה תצוגה בלבד של פעולות הלוגיקה. בהמשך ה GUI יכניס אירועים שונים שקורים ע"י הבקר במגדל (שריפה, מסלול חסום, מטוס הכריז חירום וכו) והמערכת תנסה להתמודד.

GUI

WCF Services

Logic

Server (exe)

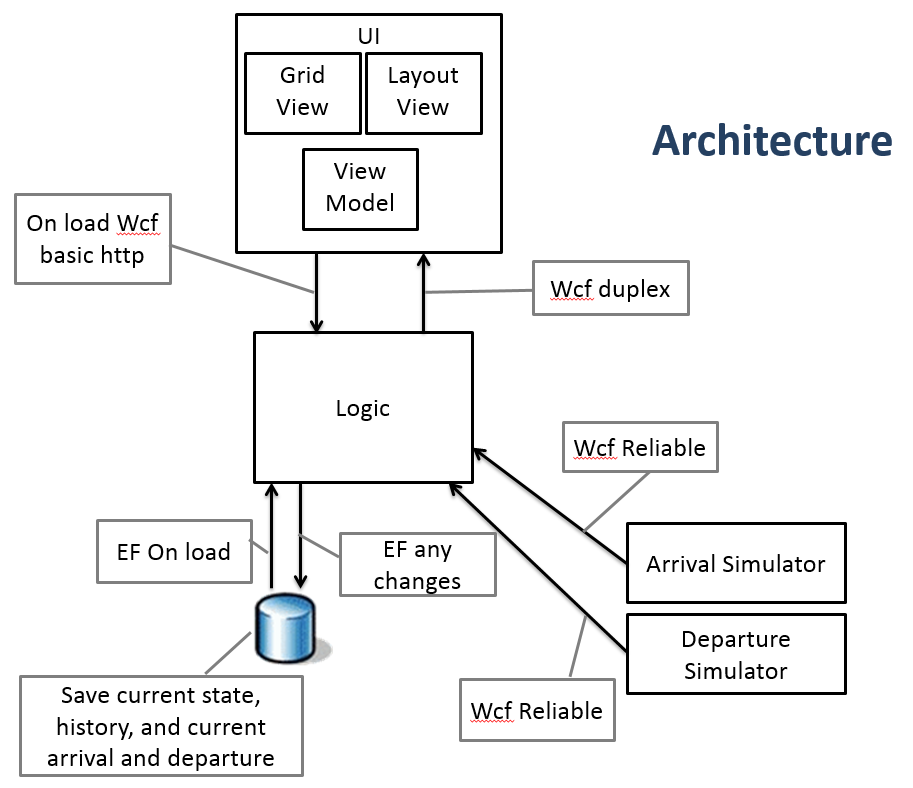
Logical Entities

Data (DB)

Simulators

Landing Simulator

Takeoff Simulator



**Scenario פתיחת אפלקציה :**

**1.הclient פותח את האפלקציה וכתוצאה מכך נפתח מסך נווט ומסך "flight control" ונשלחת בקשת login לסרבר**

**2.הסרבר מקבל את הבקשה ושולף את הנתונים של שדה התעופה מהDB**

**3.הסרבר שולח response חזרה לclient המכיל את הנתונים**

**4.****הclient מרנדר את הresponse לאיור מתאים על מסך הGUI ("flight control")**

**5.הclient שומר את המידע לגבי הסטוריית המראות ונחיתות ושולף אותם** **כאשר היוזר מחליט להחליף בעזרת חלון הנווט למסך "history flight"**

**Scenario שינוי בשדה התעופה :**

**1.הסימולטור ממציא מטוס לנחיתה\המראה או מטוס קיים משנה את מיקומו הנוכחי**

**2.הסרבר שולח callback באמצעות wcf service message לclient לגבי השינוי שהתרחש**

**3. הclient מרנדר את הresponse לאיור מתאים על מסך הGUI ("flight control")**

**4.הclient שומר את המידע לגבי נחיתה או המראה ושולף אותה כאשר היוזר מחליט להחליף בעזרת חלון הנווט למסך "history flight"**