הצעת פרויקט – י"ד הנדסת תוכנה

**שם הסטודנט:** עידו סבן

**ת.ז. הסטודנט:** 212619787

**סמל מוסד:** 570077

**שם המוסד:** מכללת אורט סינגאלובסקי

**שם רכז המגמה:** שחר אוחנה

**שם המנחה המקצועי:** סרן אופק אוחיון

**שם הפרויקט:** Skyline CRS

# תיאור הפרויקט

**מערכת הזמנות ממוחשבת** (Computer/Central Reservation System או (CRS היא מערכת שמטרתה לנהל את ההזמנות והעסקאות של ארגון, כמו חברות טיסה, מלונות, וכדומה. מערכות אלו קיימות כיום בכל מקום, בין אם ארגונים גדולים כמו חברות טיסה בינלאומיות ובין אם בתי קולנוע מקומיים.

מערכות הזמנות ממוחשבות הופיעו לראשונה בקרב חברות התעופה, שחיפשו פתרון ממוחשב על מנת לייעל את תהליך ההזמנות לטיסה. בעקבותיהן הלכו גם המלונות ושירותים דומים, עד שהפך הדבר לנפוץ כפי שהוא היום.

על מערכות אלו להיות בעלות זמינות גבוהה (high availably) עם סובלנות גבוהה לתקלות (fault tolerance), ולכם כך על המערכת להיות סקלבילית (scalable) על פי דרישה לשירותיה ולעומסים.

**Skyline CRS** היא מערכת לניהול טיסות עבור חברת תעופה בדיונית Skyline שמטרתה לנהל את כל הקשור לתהליך ההזמנה של הטיסה והעליה אליה. המערכת תעוצב בארכיטקטורת microservices על מנת לאפשר horizontal scaling של המערכת ובכך לממש את הדרישות של זמינות גבוהה, סובלנות לתקלות וסקלביליות.

המערכת תציע בנוסף ממשק משתמש אינטרנטי ללקוחות, היתקשר עם ה-API של המערכת, דרכו ניתן יהיה לחפש טיסות, להזמין טיסות (booking) ולבצע online check-in. יש לציין שלמערכת יהיה API ציבורי, כך שגם סוכנויות נסיעות חיצוניות (Online Travel Agency או OTA), לדוגמת Booking.com, Skyscanner, וכדומה, יוכלו לגשת אליו.

חשוב לציין **שלא אתייחס** בפרויקט לנושא הכבודה לטיסה וכן לעניינים כלכליים (כמו תשלום על הזמנות), אלא אתמקד בנושאי הזמנות הכרטיסים למקומות בטיסה וכל הכרוך בכך.

# רקע תיאורטי

## תהליך הזמנת הטיסות

Diagram

Description automatically generated

שלבים בתהליך:

1. **חיפוש טיסות באתר חברת התעופה** – המשתמש בוחר שדה תעופה למוצא ושדה תעופה של היעד עבור הטיסה שהוא מחפש בין טווח תאריכים שהוא בוחר. כמו כן הוא יכול לבחור גם במספר הנוסעים לטיסה. לאחר שהוא מבצע חיפוש, תוצג לו רשימת טיסות מתאימות לפרמטרים שנתן. מהן יוכל המשתמש לבחור אחת מהטיסות המוצעות ולבצע הזמנה.
2. **הזמנת הטיסה** – לאחר בחירת הטיסה מרשימת הטיסות, המשתמש ימלא פרטים אישיים, ולאחר מכן אף יוכל לבחור במקומות בטיסה. לאחר שיבצע את ההזמנה הוא יקבל מספר הזמנה (PNR locator או Booking number) המזהה את הזמנתו. באמצעות מספר ההזמנה יוכל המשתמש לבדוק בכל עת את מצב ההזמנה וכן לגשת לכרטיס ה-e-ticket שיקבל במייל גם דרך האתר. כעת ההזמנה מחכה לאישור ותישאר במצב זה עד שישלח לנוסע כרטיס דיגיטלי.
3. כרטיס הטיסה הדיגיטלי (e-ticket) ישלח לנוסע באימייל לאחר כמה זמן (לרוב עד 3 ימים ברוב חברות הטיסה), שם יהיו פרטי ההזמנה השונים. רק לאחר שהתקבל כרטיס הטיסה ניתן יהיה לעשות צ'ק-אין.
4. **צ'ק-אין מקוון (online check-in)** – הצ'ק-אין לטיסה נפתח 72 שעות לפני זמן העלייה למטוס. תהליך זה הכרחי על מנת לקבל את כרטיס העלייה למטוס (boarding pass) – בלעדיו לא יתאפשר לעלות על הטיסה! בתהליך הצ'ק-אין על הנוסעים יהיה למלא פרטים חסרים (כמו בחירת מקומות במקרה שלא בחרו בשלב ההזמנה) ולענות על שאלות אבטחה שונות. את תהליך הצ'ק-אין המקוון יבצעו הנוסעים דרך אתר חברת התעופה, על ידי הכנסת מספר ההזמנה ומספר פרטי אימות נוספים, ובתומו יקבלו את כרטיס העלייה למטוס דרך האימייל והאתר.
5. **עליה למטוס** – הנוסעים יגיעו לשדה התעופה עם כרטיס העלייה למטוס שקיבלו (דיגיטלי או מודפס), יבצעו את ההליכים הדרושים בשדה, ויעלו על הטיסה בזמן שצוין בכרטיס העלייה למטוס.

## מונחים

* **Passenger Name Record (PNR)** – קובץ הנוצר עבור כל הזמנה, המתאר את פרטי כל הנוסעים וכן פרטים נוספים (כמו פרטי התקשרות, מצב ההזמנה, וכדומה). לאור זאת, ניתן לומר ש-PNR הוא מונח טכני המקביל להזמנה (booking).
* **PNR Locator (Booking Number)** – מספר המזהה את ה-PNR שנוצר עבור ההזמנה. מספר זה הכרחי על מנת לבדוק את מצב ההזמנה, לבצע שינויים בהזמנה, לבצע צ'ק-אין לטיסה, וכדומה.
* **E-ticket** – כרטיס (מסמך) דיגיטלי המתקבל לאחר אישור ההזמנה (במציאות לאחר בדיקת פרטים וחיוב האשראי) המתאר את ההזמנה ופרטיה. בין הפרטים העיקריים שמכיל ה-e-ticket הוא מספר ההזמנה (PNR locator) שיאפשר לבצע check-in בשלב מאוחר יותר.
* **צ'ק-אין (Check-in)** – שלב שניתן לביצוע או באופן מקוון (online check-in), לרוב בין 24-72 שעות לפני הטיסה, או בשדה התעופה לפני העלייה למטוס. בשלב זה נבדקים הפרטים האישיים של הנוסעים (למשל מספר דרכון, תאריך לידה, וכולי) לצורכי אימות, וכן ניתנת האפשרות לנוסעים לרוב לבחור מקומות (עבור תשלום נוסף), להגדיל את כמות הכבודה, להוסיף אוכל לטיסה, וכולי. לבסוף, יונפק עבור על נוסע כרטיס עליה למטוס (boarding pass). למעשה ללא שלב זה לא ניתן יהיה לקבל את כרטיס העלייה למטוס, וכן, לא יהיה ניתן לעלות לטיסה.
* **כרטיס עליה למטוס (Boarding Pass)** – הכרטיס באמצעותו נוסע עולה למטוס. הכרטיס יכיל פרטים כמו מושב בטיסה, זמן העלייה למטוס, פרטי נוסע, טרמינל, וכדומה. כרטיס זה יכול להיות או דיגיטלי על ידי ביצוע צ'ק-אין מקוון או מודפס אם בוצע בצ'ק-אין בשדה התעופה.

# תהליכים עיקריים בפרויקט

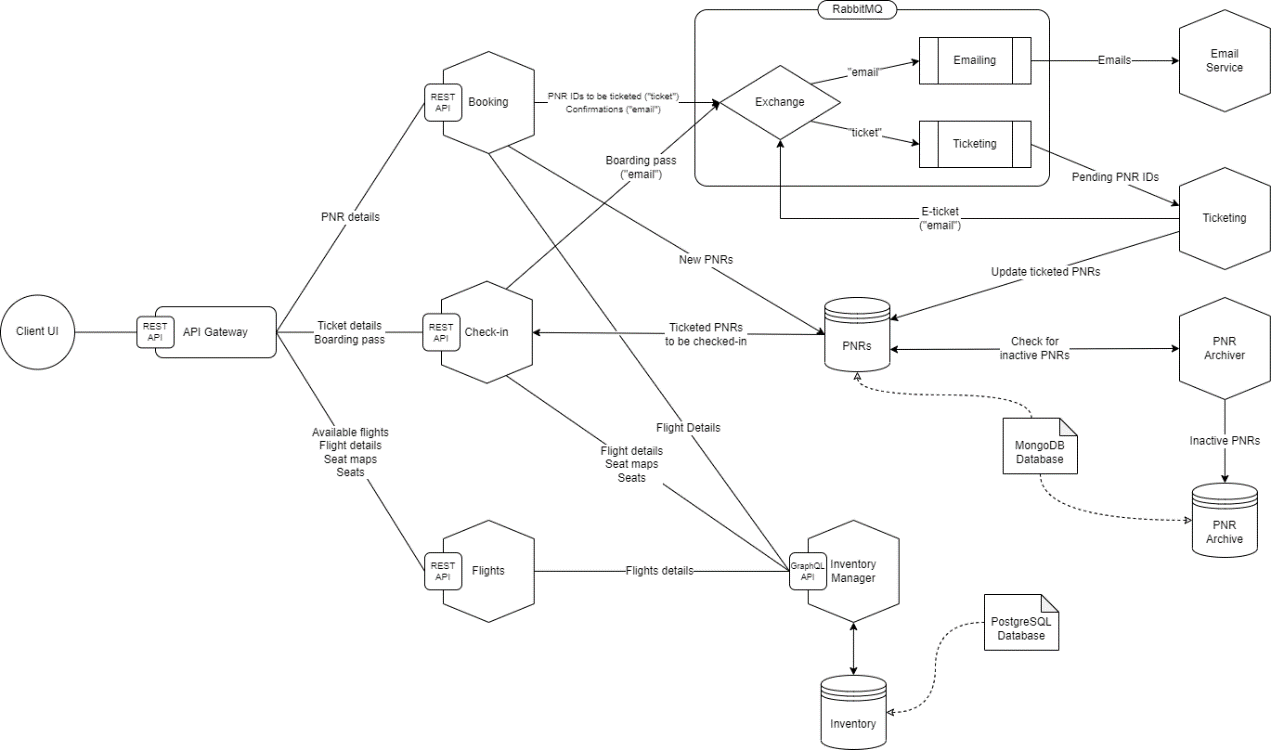
* חיפוש טיסות על פי פרמטרים נתונים (מוצא, יעד, טווח תאריכים, מספר נוסעים, מחלקת טיסה).
* הצגת פרטי טיסה (פרטי הטיסה, מבנה המטוס, מקומות פנויים).
* הזמנת טיסה (יצירת PNR).
* הצגת פרטי הזמנה.
* עדכון או ביטול הזמנה.
* ביצוע צ'ק-אין לטיסה וקבלת כרטיס עליה למטוס..
* שליחת אימיילים (e-ticket, כרטיס עליה למטוס, אישורי הזמנה וביטול).
* ייעול מסד הנתונים של ה-PNR-ים על ידי העברת PNR-ים לא פעילים לארכיון.

# טכנולוגיות בפרויקט

* **Docker** – פלטפורמה המאפשרת הרצת קונטיינרים (containers) ובניית תמונות (images) ליצירתם. בפרויקט, כל microservice ייארז כתמונה נפרדת וירוצו בתוך קונטיינרים נפרדים המתקשרים זה עם זה על גבי הרשת.
* **Kubernetes** – מערכת שפותחה על ידי גוגל המשתמשת לניהול ופריסה אוטומטית של קונטיינרים (אורכסטרציה של קונטיינרים – container orchestration). למעשה היא זו שתאפשר לקשר בין כל חלקי הפרויקט השונים ותעזור לממש את עקרונות הזמינות הגבוהה, סובלנות לתקלות וסקלביליות.
* **RabbitMQ** – פלטפורמת תיווך הודעות (message broker) המשמשת לניהול תורים תחת פרוטוקולים שונים, בעיקרAdvanced Message Queuing Protocol (AMQP). מטרת השימוש במתווך הודעות היא מניעת עומסים על ה-backend על ידי דחיית עבודות שלוקחות זמן ובכך לאפשר זמינות גבוהה יותר של המערכת ומניעת שגיאות במקרה של תקלות במערכת.
* **PostgreSQL** – מערכת לניהול בסיסי נתונים רציונליים (RDBMS). בפרויקט תשמש לאחסן את נתוני הטיסות והמקומות בהן.
* **Hasura** – מוצר שיוצר הפשטה על גבי בסיס נתונים רציונאלי, כמו PostgreSQL, וחושף באופן אוטומטי ממשק GraphQL על פי סכמת מסד הנתונים. ישמש להפשטת ההתממשקות עם מסד הנתונים של הטיסות וכן לשליטה על הגישה אליו.
* **MongoDB** – מערכת לניהול בסיסי נתונים לא רציונליים (NoSQL database) המשתמשת במסמכים (documents) דמויי JSON לאחסון הנתונים באוספים (collections), בניגוד למסדי נתונים רציונליים המשתמשים בטבלאות לייצוג הנתונים. בפרויקט תשמש לאחסן את ה-PNR-ים של ההזמנות השונות, שלרוב מאוחסנים כקבצים נפרדים, שכן אופי הנתונים תואם לאופי אחסונם במערכת זו.
* **Git** – מערכת לניהול גרסאות מבוזרת (distributed VCS) הפופולרית בעולם. תשתמש לנהל את הגרסאות השונות במהלך הפיתוח של הפרויקט.
* **GitHub** – שירות ניהול ואחסון אינטרנטי עבור Git repositories. יחד עם Git היא תשמש לניהול הקוד של הפרויקט לאורך הפיתוח.
* **ספריות עבור ה-backend לשפה Python:**
  + **FastAPI** – Web framework אסינכרונית ליצירת REST APIs. תשתמש בתוכנה Uvicorn להרצת השרת. ספרייה זו תשמש ליצירת microservices החושפים REST API במערכת.
  + **aio-pika** – ספרייה להתממשקות עם RabbitMQ. תשמש microservices שצריכים לשלוח הודעות לתורים או לקבל מהם הודעות.
  + **Strawberry** – ספרייה לתקשורת עם ממשקי GraphQL ויצירתם. תשמש microservices לתקשורת עם Hausra.
  + **Motor** – ספריית ODM לתקשורת אסינכרונית עם MongoDB. תשמש microservices שצריכים להתממשק עם מסד הנתונים של ה-PNR-ים.
* **ספריות עבור ה-backend לשפות JavaScript/Typescript:**
  + **Express** – ספרייה ליצירת שרת web אסינכרוני החושף REST API**.** ספרייה זו תשמש ליצירת microservices החושפים REST API במערכת.
  + **amqplib** – ספרייה להתממשקות עם RabbitMQ. תשמש microservices שצריכים לשלוח הודעות לתורים או לקבל מהם הודעות.
  + **graphql-request** – ספרייה לתקשורת עם ממשקי GraphQL ויצירתם. תשמש microservices לתקשורת עם Hausra.
  + **Mongoose** – ספרייה ODM לתקשורת עם MongoDB. תשמש microservices שצריכים להתממשק עם מסד הנתונים של ה-PNR-ים.
  + **Nodemailer** – ספרייה לשליחת אימיילים, על גבי פרוטוקול SMTP ופרוטוקולים נתמכים אחרים. תשמש את Email Service.
* **ספריות עבור ה-frontend לשפות JavaScript/Typescript:**
  + **React** – ספרייה ליצירת אפליקציות web שפותחה על ידי פייסבוק. הספרייה תשמש ליצירת אתר חברת התעופה.
  + **Axios** – ספריית JavaScript/Typescriptמבוססת Promises המפשיטה יצירת בקשות HTTP. תשמש לתקשורת בין ממשק המשתמש ל-API של המערכת.

# תרשימים

## תרשים המערכת



## תרשים מסכים (User Flow)

Diagram

Description automatically generated

# לוחות זמנים



# אישורים

**חתימת הסטודנט: A picture containing text, clipart

Description automatically generated**

**חתימת רכז המגמה:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**אישור ממשרד החינוך:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_