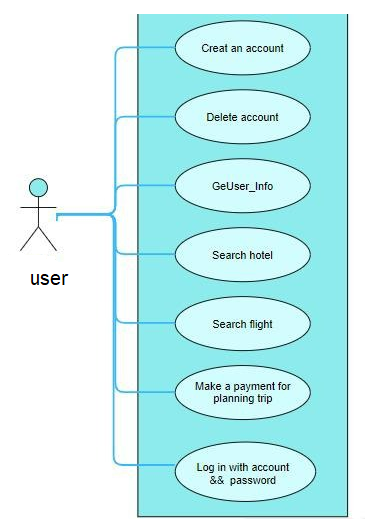
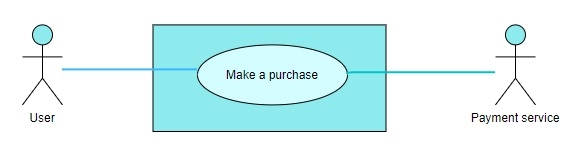
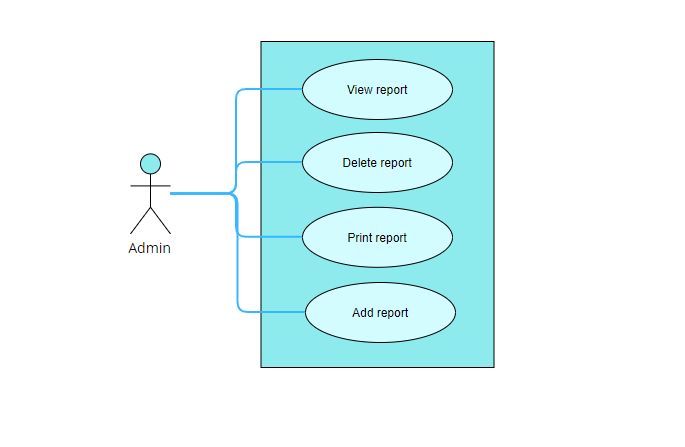
****

**BUILD YOUR TRIP**

Use case diagram

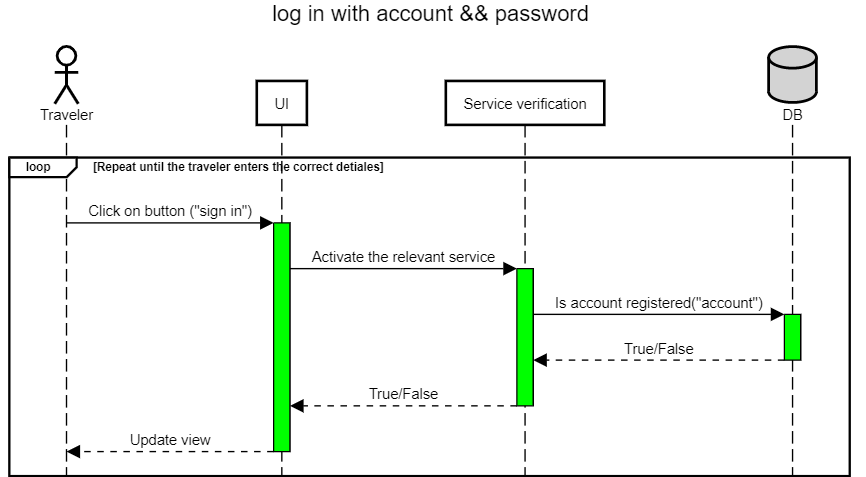


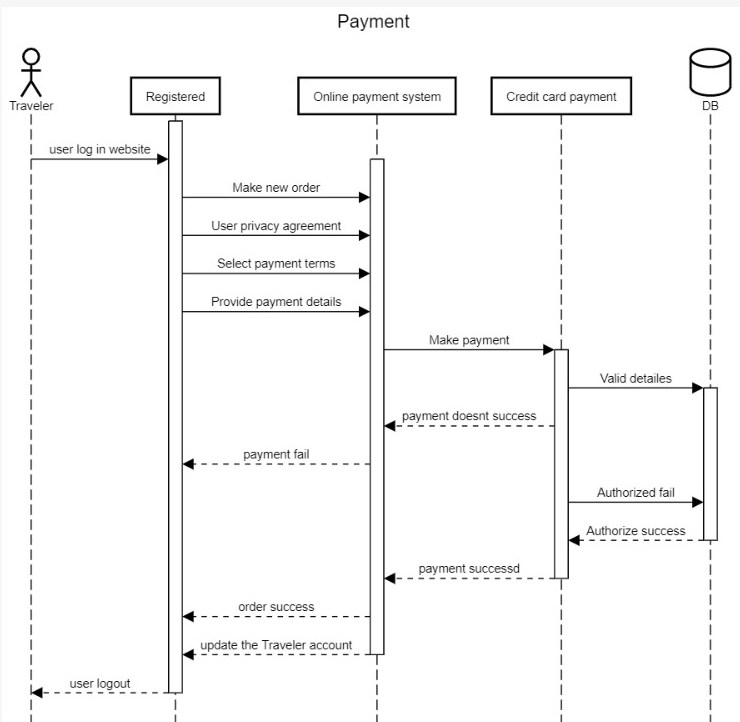


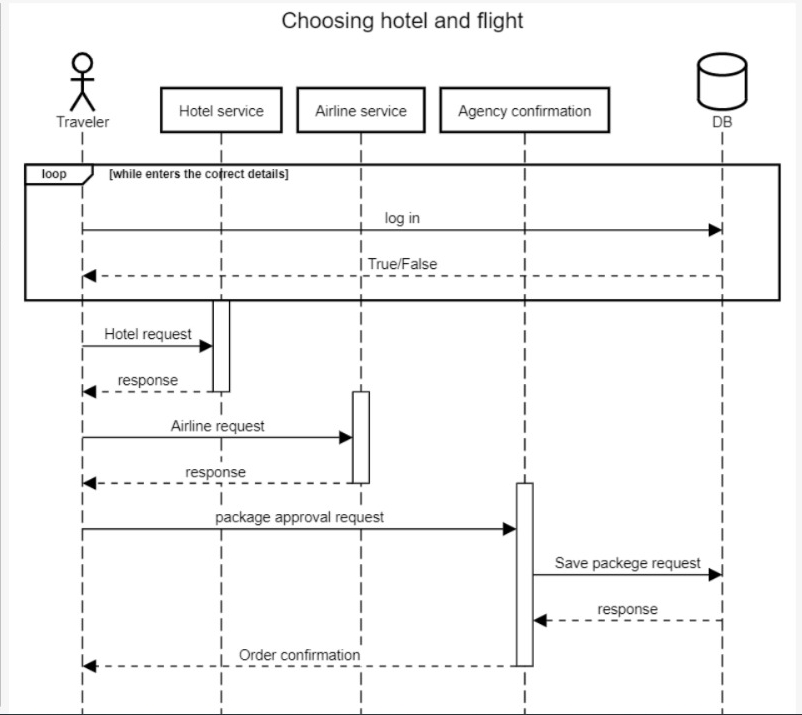


Sequence diagrams

Log in account and password:



Make a payment:

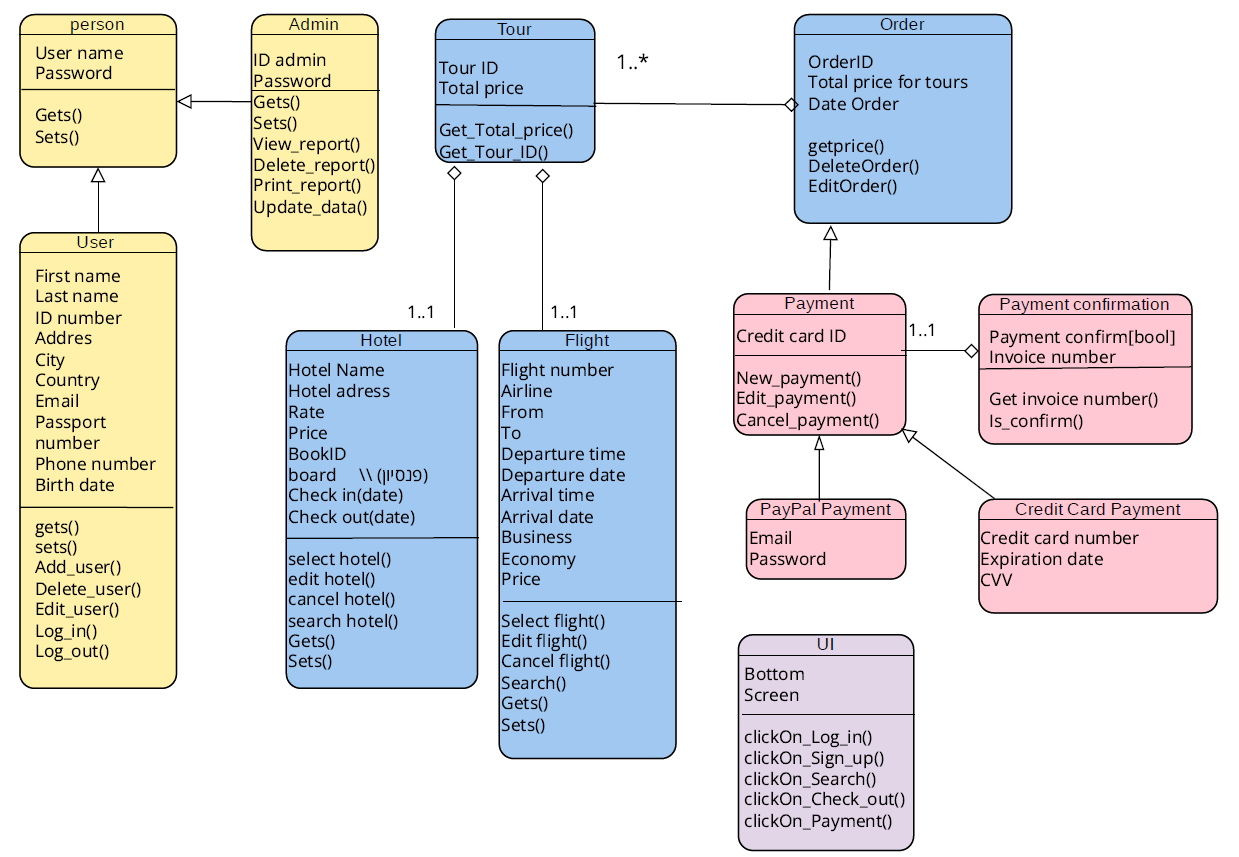
Choosing hotel and flight:

Class diagram

הערה :

**Tour**מכיל מלון + טיסה -

**Order**יכול להכיל כמה חבילות כלומר כמה מלונות + טיסות -



[**Architectural Pattern**](#_tvm2gsejkijx)

מודל מורכב משלוש שכבות **MVC**

Model: Data logic.  
controller: processor. Control and decides how data is displayed his job is to process.

View: represents current model state.

במקרה שלנו הגישה אל המערכת נעשית באמצעות דפדפן אינטרנט, ומרבית השירותים מוצעים ע"י שרתים שונים. לכן יש צורך להשתמש בארכיטקטורת **לקוח- שרת** בנוסף למודל ה-MVC.

**דוגמא**:   
מלחיצה של המשתמש על התחברות Log-in מגיעים לUSER ומתחברים עם שם משתמש וסיסמא .  
ה-CONTROLLER לוקח את הנתונים ושולח אותן ל - MODEL כדי לבדוק אם יש אישור ואכן המשתמש קיים במערכת. לבסוף הMODEL - מחזיר לCONTROLLER - שהמשתמש קיים ומתבצעת התחברות.

ברגע שהMODEL מאשר את ההתחברות המשתמש אז יש התחברות לשרת SERVER.  
הclient מכיל בתוכו את ה VIEW, MODEL, CONTROLLER .

**SERVER**

**CLIENT**

**controller**

**GetHotel()**

**GetFlight()**

**UpdateView()**

**VIEW**

**UI**

**MODEL**

**User**

**Hotel**

**Flight**

**Tour**

**Payment**