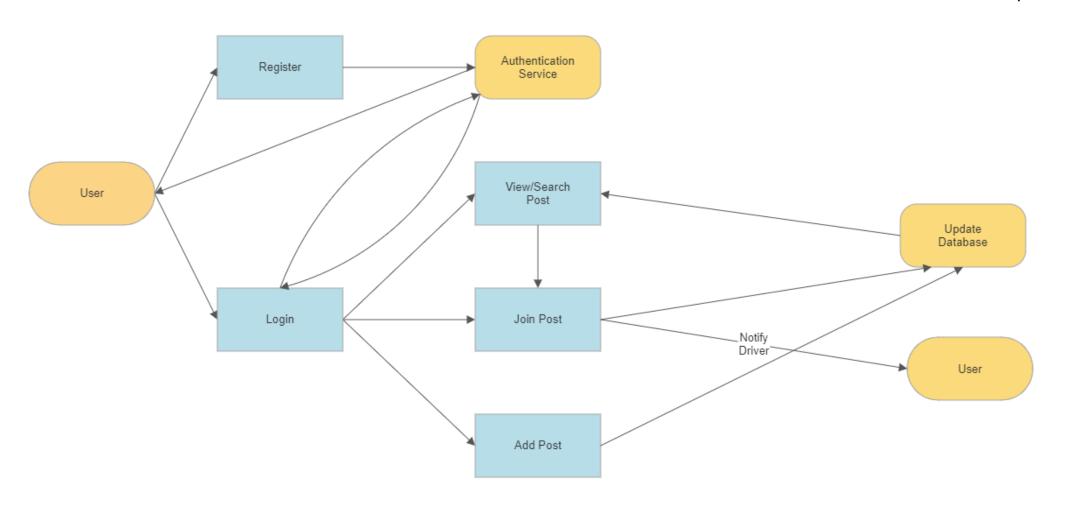
# הנדסת תוכנה מטלה 3 – מסמך ניתוח

206284960 – ליאל ביטון – 207304353 – מוריה בראל, 206284960 – חי לוי, 215021923 – ליאל ביטון

# **Use Case Diagram:**

שחקנים בצבע צהוב / תהליכים בצבע תכלת



# **Class Diagram:**

	<u>DateAndTimeFormat</u>		<u>PostAdapter</u>	<u>UsersAdapt</u>	'er			
Adapters	> getDateAndTime() : public > compareDateAndTime() : public	databaseReference : DatabaseReference     usedByHomeFrg : Boolean  > getView() : public     setupPostText() : public     addPersonsDrawings() : private     setupJoinRideBtn() : private		> getItem() : pu > getView() : pu				
AdminActivities	AddCity  cityInput: TextInputLayout priceInput: TextInputLayout mapView: MapView geocoder: Geocoder submitBtn: Button databaseReference: DatabaseReferer insertedCity: String insertedPrice: String  getView(): protected > setupSubmitButton(): public > checkCityExistsInDatabase(): private	<ul> <li>autoCompleteTextViewCity: AutoComplete adapterCities: ArrayAdapter<string> <ul> <li>eiw oder</li> <li>citiesAndPrices: HashMap<string, li="" string<=""> <li>citiesList: ArrayList<string></string></li></string,></li></ul></string></li></ul>		ring>	• removeUserBtn : ImageButton • pricesBtn : ImageButton • pricesBtn : ImageButton • addCityBtn : ImageButton • userName : TextView • email : TextView > onCreate() : protected > setupPricesBtn() : public > setupRemoveUserBtn() : private > initProfile() : private	listVii users users datal autof  onCr setup setup remo	RemoveUser  searchView: SearchView listView: ListView usersList: ArrayList <user> usersAdapter: UsersAdapter databaseReference: DatabaseReference autoProfile: FirebaseAuth  onCreate(): protected setupSearchView(): public setupListView(): private setupItemFromListView(): private removeUserFromAuthDB(): private initUsersList(): private</user>	
Fragments	AddPostFragment.  autoCompleteTextViewSrc:AutoCompleteTextView autoCompleteTextViewDest:AutoCompleteTextView adapterSrcCities:ArrayAdapter <string> adapterDestCities:ArrayAdapter<string> databaseReference:DatabaseReference cities:ArrayList<string> datePickerDialog:DatePickerDialog dateButton:Button bour,minute:int,int submitButton:Button citiesAndPrices:HashMap<string,string> source, destination, seats, description:private String date, time, cost:private String  onCreateView():public setupSubmitButton():private insertPostToDataBase():private setupTimeButton():private setupTimeButton():private setupTemButton():private setupTemButton():private setupTemButton():private setupTemButton():private setupTemButton():private setupTemAndToAutoComplete():private makeDateString():private getMonthFormat():private initCitiesList():private</string,string></string></string></string>		postAdapter: PostAdapter     valueEventListener: ValueEventListener     databaseReference: DatabaseReference     onCreateView(): public     setupSearchView(): private     setupListView(): private     initPostList(): private     onDestroy(): public		MyPostsFragment  • databaseReference : DatabaseReference • active_rides : ListView • ride_history : ListView • HistoryRideAdapter : PostAdapter • ActiveRideAdapter : PostAdapter • IistTitle : TextView  > onCreateView() : public > setupListView() : private > setupSwapButton() : private > initPostList() : private > classificationHistoryOrActive() : private		AddPostFragment  • profileName : TextView • profileEmail : TextView • changePassword : TextView • changeEmail : TextView • adminBtn : Button  > onCreateView() : public > setAdminBtn() : private > setChangeBtns() : private > initProfile() : private > classificationHistoryOrActive() : private	

### **More Classes:**

#### Post User postID : String publisherID : String id : String firstname : String publisherFirstName : String lastname : String publisherLastName : String email : String seats : String Uid : String isAdmin : Boolean · source : String destination : String leavingTime : String > getUid() : public > getIsAdmin() : public leavingDate : String cost : String > getFirstname() : public > getLastname() : public · description : String passengerIDs : List<String> > getEmail() : public > getId(): public > getPublisherID() : public > getFullName() : public > getLeavingDate(): public > toString(): public Objects > getPostID() : public > getDescription() : public > getSource() : public > getDestination() : public > getPublisherFirstName() : public > getPublisherLastName() : public > getCost() : public > getSeats() : public > getAvailableSeats() : public > getLeavingTime() : public > getPassengerlDs() : public > addPassenger() : public > getPublisherFullName() : public > isFull() : public > toString() : public > createPost() : public AddPost ForgotPassword AddPost autoCompleteTextViewSrc : AutoCompleteTextView emailTxt : String databaseReference : DatabaseReference autoCompleteTextViewDest : AutoCompleteTextView idTxt : String authProfile : FirebaseAuth adapterSrcCities : ArrayAdapter<String> • databaseReference : DatabaseReference firstnameTxt : String adapterDestCities : ArrayAdapter<String> lastnameTxt : String databaseReference : DatabaseReference > setupSendNewPasswordButton() : private emailTxt : String cities : ArrayList<String> > checkIdAndEmailOnDatabase(): private idTxt : String datePickerDialog : DatePickerDialog > setupBackBtn() : private passwordTxt : String dateButton : Button > resetPassword(): private confirmPasswordTxt : String timeButton : Button hour : int > onCreate(): protected minute : int > setupSignUpButton(): private submitButton : Button > setupBackToLoginBtn() : private Login citiesAndPrices : HashMap<String, String> > authRegistration(): private source : String > realTimeRegistration(): private · autoProfile : FirebaseAuth destination : String > checkAdmin() : private user: User seats : Stringdescription : String UsersActivities firebase url : String databaseReference : DatabaseReference date : String time : String > setupLoginButton(): private cost : String > setupRegisterButton() : private > setupForgotPasswordButton(): private > onCreate(): protected > getIdByEmail(): private > setupSubmitButton() : private > str to boolean(): private > checkValidTime() : private > insertPostToDataBase() : private > setupTimeButton() : private > setupDateButton() : private > setupFromAndToAutoComplete(): private MainActivity > initDatePicker() : private > makeDateString() : private bottomNavigationView : BottomNavigationView > getMonthFormat() : private selectedFragment : Fragment initCitiesList() : private > setupNavBar() : private

### **Object Diagram:**

#### <u>Post</u>

- · postID : String
- publisherID : String
- publisherFirstName : String
- publisherLastName : String
- seats : String
- source : String
- destination : String
- leavingTime : String
- leavingDate : String
- cost : String
- description : String
- passengerIDs : List<String>
- > getPublisherID(): public
- > getLeavingDate() : public
- > getPostID(): public
- > getDescription(): public
- > getSource() : public
- > getDestination() : public
- » getPublisherFirstName() : public
- > getPublisherLastName(): public
- > getCost() : public
- > getSeats(): public
- > getAvailableSeats() : public
- > getLeavingTime(): public
- > getPassengerIDs(): public
- > addPassenger() : public
- > getPublisherFullName() : public
- > isFull(): public
- > toString(): public
- > createPost() : public

#### User

- id : String
- firstname : String
- lastname : String
- email : StringUid : String
- isAdmin : Boolean
- > getUid(): public
- > getIsAdmin(): public
- > getFirstname(): public
- > getLastname() : public
- > getEmail() : public
- > getld() : public
- > getFullName() : public
- > toString(): public

#### <u>DateAndTimeFormat</u>

- > getDateAndTime() : public
- > compareDateAndTime(): public

#### <u>UsersAdapter</u>

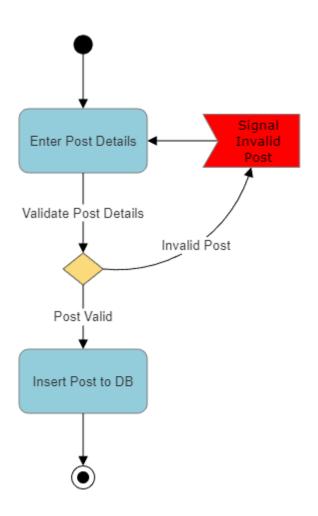
- > getItem(): public
- > getView(): public

#### PostAdapter

- databaseReference : DatabaseReference
- usedByHomeFrg: Boolean
- > getView(): public
- > setupPostText() : public
- > addPersonsDrawings() : private
- > setupJoinRideBtn() : private

### Objects

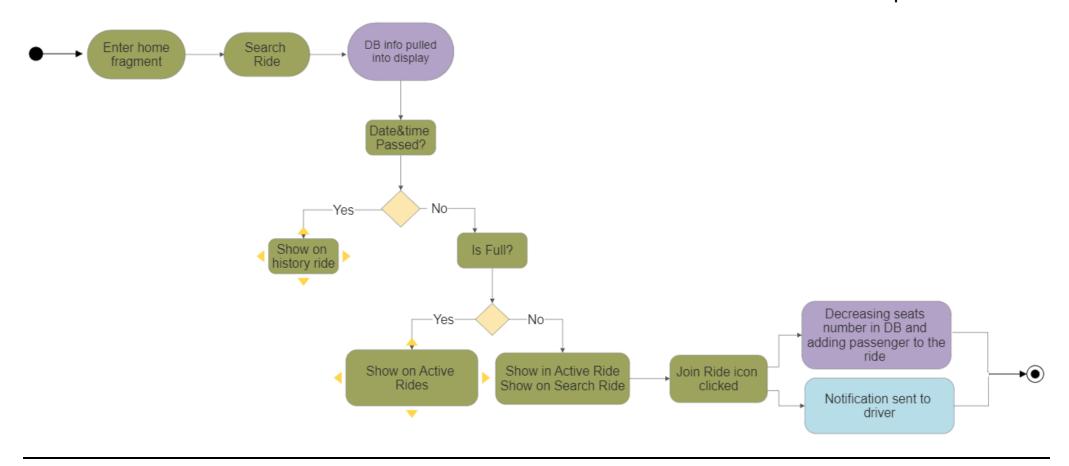
# **Activity Diagram:**



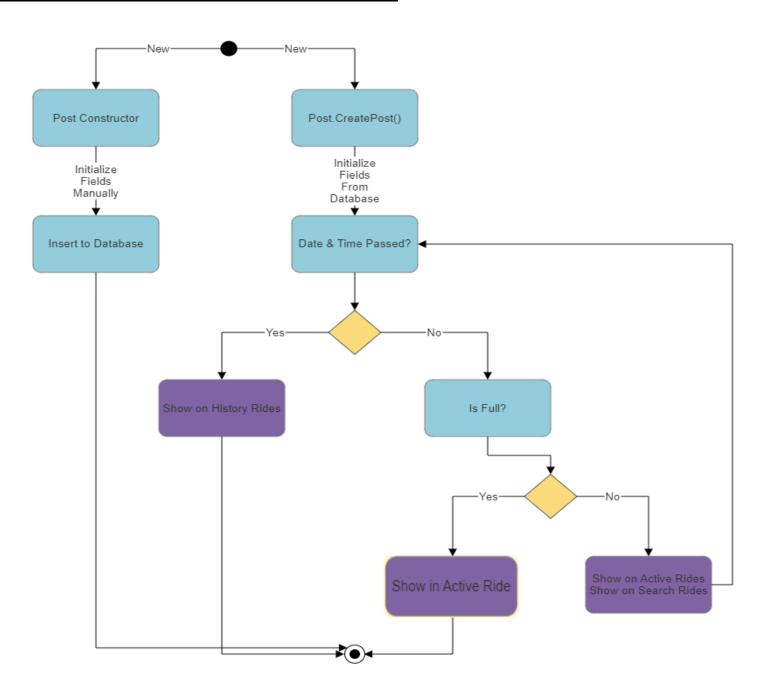
דיאגרמה של תהליך הוספת פוסט על ידי נהג

## **Sequence Diagram:**

דיאגרמה של תהליך הוספת חיפוש נסיעה והצטרפות אליה על ידי הטרמפיסט



# **State Machine Diagram for 'Post' Object:**



## **ERD**:

