DiveRank class

// Properties \\

1. Name – String
2. Description – String
3. DivesNeeded – int

DiveRankGiven class

// Properties \\

1. DateGive – Data Time
2. ClubGiven – DivingClub Object
3. Certificate – PICTURE

Item class

// Properties \\

1. Name – String
2. ID – String
3. Type – Enum

Diver Items class

// Properties \\

1. Item – Item Object
2. Amount – int
3. Description – String

Address class

// Properties \\

1. **Street Name** – String (e.g., "Main Street")
2. **House/Building Number** – String (Some addresses include letters, e.g., "12B")
3. **City** – String (e.g., "New York")
4. **State/Province/Region** – String (e.g., "California" or "Ontario")
5. **Postal/ZIP Code** – String (e.g., "10001" or "W1A 1AA")
6. **Country** – String (Ensure it's validated using an API)

Country class

// Properties \\ // Use API for checking if it’s valid?

1. Name – String (Setter Use API to check if country is real)
2. Diving regulations - DivingRegulations Object
3. Languages – Linked List just for fun!

Person class

// Properties \\

1. ID – String ( validate ID from function I made )
2. First Name – String ( REGEX)
3. Last Name – String ( REGEX)
4. Date of Birth – String (Validate before saving)

Diver class : Person

// Properties \\

1. ID – String ( : person)
2. First Name – String ( : person)
3. Last Name – String ( : person)
4. Date of Birth – String ( : person)
5. Unique diver ID – String ( Make something up )
6. Dives done – int (shows how maby dives the diver did)
7. CurrentRank – כוכבים – DiveRankGiven Object
8. PrevRanks[] – כוכבים קודמים – DiveRankGiven[] array Object
9. Diving log => getting services from diving clubs (needs to be kept well can be destroyed easly)

Diver Instructor class : Diver

// Diving Instructors will approve and document the dives they are doing

// Properties \\

1. ID – String ( : person)
2. First Name – String ( : person)
3. Last Name – String ( : person)
4. Date of Birth – String ( : person)
5. Unique diver ID – String ( : diver )
6. Dives done – int ( : diver )
7. Start working – Date
8. End working – Date
9. Current Working Clubs[] – DivingClub[] Object //Can work in a few clubs at the same time
10. Clubs Worked[] – DivingClub Object array

Diving Club Class

// Properties \\

1. License – String ( REGEX ) (MUST BE UNIQUE)
2. static string[] Licenses = [] // Will store all of the licenses that were made
3. Name – String ( REGEX)
4. Contact name – String
5. Address – Address Object ( Will include in it Country – OBJECT)
   1. Diving regulation – Inside the country -> inside the address
6. Phone number - String ( REGEX)
7. email – String ( REGEX )
8. divingSite - divingSite OBJECT
9. hasSite – Bool
10. WEBsite address ( REGEX )
11. DivingLog – DivingInfo[]

Diving Site Class : Address

// Properties \\

1. name – string
2. description – string (small desc on the site)
3. Length – double (meters)
4. Width – double (meters)
5. Deep – double (meters)
6. Water Salty – bool.

Signature Class

// Properties \\

Object – Object Object (The diver that signed) = We will use different objects depending on who signs, Because we want to enable a diver to sign as well as a diving club

DateTime – DateTime

Signature – FIND A WAY TO REPRESENT THE SIGNATURE

Diving Info Class

// Properties \\

1. Diving Club – Diving Club Object
2. Diving site – Diving Site Object
3. Dive date – date
4. HeadDiver = diver instruction class (The one who will fill the information)
5. HeadDiver Works currently at club? BOOL (must be true)
6. Diver Instructors[] – Array of diver instruction class
7. Divers[] - Array of diver class
8. Item List[] – List KeyValue (string – diver name, value – Diver Item[])
9. Water entre – date time
10. Water exit – date time
11. Divers Signatures[] – Array of diver signutures
12. Diving club Signature – 1 Signature of the diving club (not must) but need for the dive to be recognized by other diving clubs
13. Water temperature – double
14. Tide level double (0-1) 1 = high, 0 = low, in the middle will be in between, will be shown in %
15. Was dive valid – BOOL (will be set after all data input based on)
    1. diving regulations – ALL VALID?
    2. Head diver works currently?
    3. All divers signed?
    4. Is the diver that submitted the information is a head diver? (DIVER RANK CHECK)

**Information:**

* Diving is usually in pairs
* Hard to find a partner
* Not all needed equipment is always available

Diving **MUST**

* Diving journey will happen only when in a diving club (else show error!)
  + Need ability to add diving clubs!!
* At the end of the dive, all divers must sign that they have been diving using a digital pen

**SYSTEM SCREENS**

To log in to system

// Dictionary that will contain all of those details

Username == Email.

Password == ( REGEX: 8 digits exactly [a-z0-9] ONLY!)

Find a diving club

// Finding a club will be used by one of the following (Select box with a few options that will contain)

1. Club name
2. Club License
3. Club Contry
4. Club Address

Edit Diver Data

// Finding a club will be used by one of the following (Select box with a few options that will contain)

1. Edit the data of the diver (including rank)
2. Make a print of data ranks that will show the ranks and amount of dives made

Note: Add a pre existing DB of items, so that if a diver tries to take an item and there is none, he will get an error, FOR EVERY DIVING CLUB.