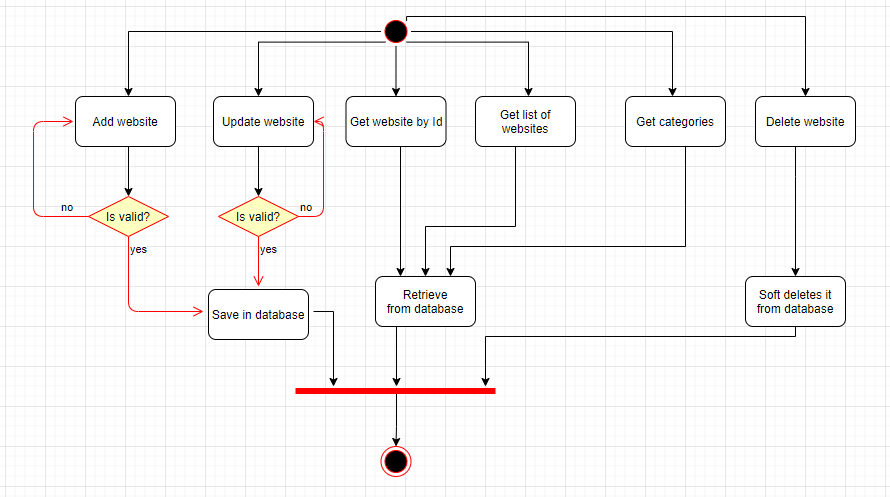
# Technologies and design patterns

* .NET Core
* Entity Framework Core (Code first).
* Singleton pattern (for IPasswordService).
* Dependency injection for easy replacement of implementations.
* Middleware, which catches exceptions and returns unified response to the client.
* Data Transfer objects for better communication with the client and hiding sensitive data.
* NUnit for unit testing the services in the Application.
* On Start up there are 2 methods called:
  + app.UpdateDatabase() -> which migrates database to the latest version.
  + app.SeedDatabase() -> which modifies the data in the database.
* Soft deleted records (Such a record should implement ISoftDeleteEntity interface) are excluded from all queries In the DataContext class with a ‘QueryFilter’.
* In order to GET deleted record you should append ‘.IgnoreQueryFilters()’ to the query.
* I use CODEs for unique identifier of each Category (as they are predefined), so all Add/Update request should pass Category Code instead of ID. This is made for easier maintenance of the application. I seed the database with five categories. It is only possible to get all categories.

|  |  |
| --- | --- |
| Code | Name |
| TECH | Technology |
| SPORT | Sport |
| EDU | Education |
| NEWS | News |
| ENT | Entertainment |

# Activity diagram



# Component Diagram

