Amplifir

Architecture and Design

Note: Until the first production release of the codebase, this document will only be versioned through Git. After the first production release, all versions of this document must be declared on the first page and copies of each version must exist on the repository.

Created By: João Pedro Martins Neves

General Guidelines

- All files **must** be saved as UTF-8.
- Internal system (server, database, etc.) DateTimes and Timestamps **must** be UTC, only presentation code should be presented as local date-times.
- The line-endings/line-breaks **should** be LF, instead of Windows CRLF, to provide better multiplatform support. .NET Core is multi-platform.
- Use **spaces**, not tabs.
- You **must** insert a space after opening and before closing non-empty parenthesis.

Solution Architecture

- UI/
  - Presentation/ (Angular code)
  - Web/ (ASP.NET Core server)
    - Controllers/
    - Filters/
    - (...)
  - Desktop/ (Electron.NET)
  - Mobile/
- Core/
  - Interfaces/
  - Exeptions/
  - DTOs/ (classes used to return data from the API to the client)
  - Models/ (classes used for internal logic)
  - Entities/ (classes that describe a database entity/tables)
  - DomainServices/ (internal services that control/manage the communication between the data access/infrastructure layer and the API. Facades).
  - Enums/
  - Utilities/
- (Core) ApplicationTypeFactory/ (a factory that provides implemented types by the infrastructure to the UI/Web layer)
- Infrastructure/
  - DataAccess/
  - Logging/
  - Services/
    - Email/
    - SMS/
- Infrastructure.DatabaseCode/ (code representing the database schema and its content)
- Tests/

Database

The database is **PostgreSQL**.

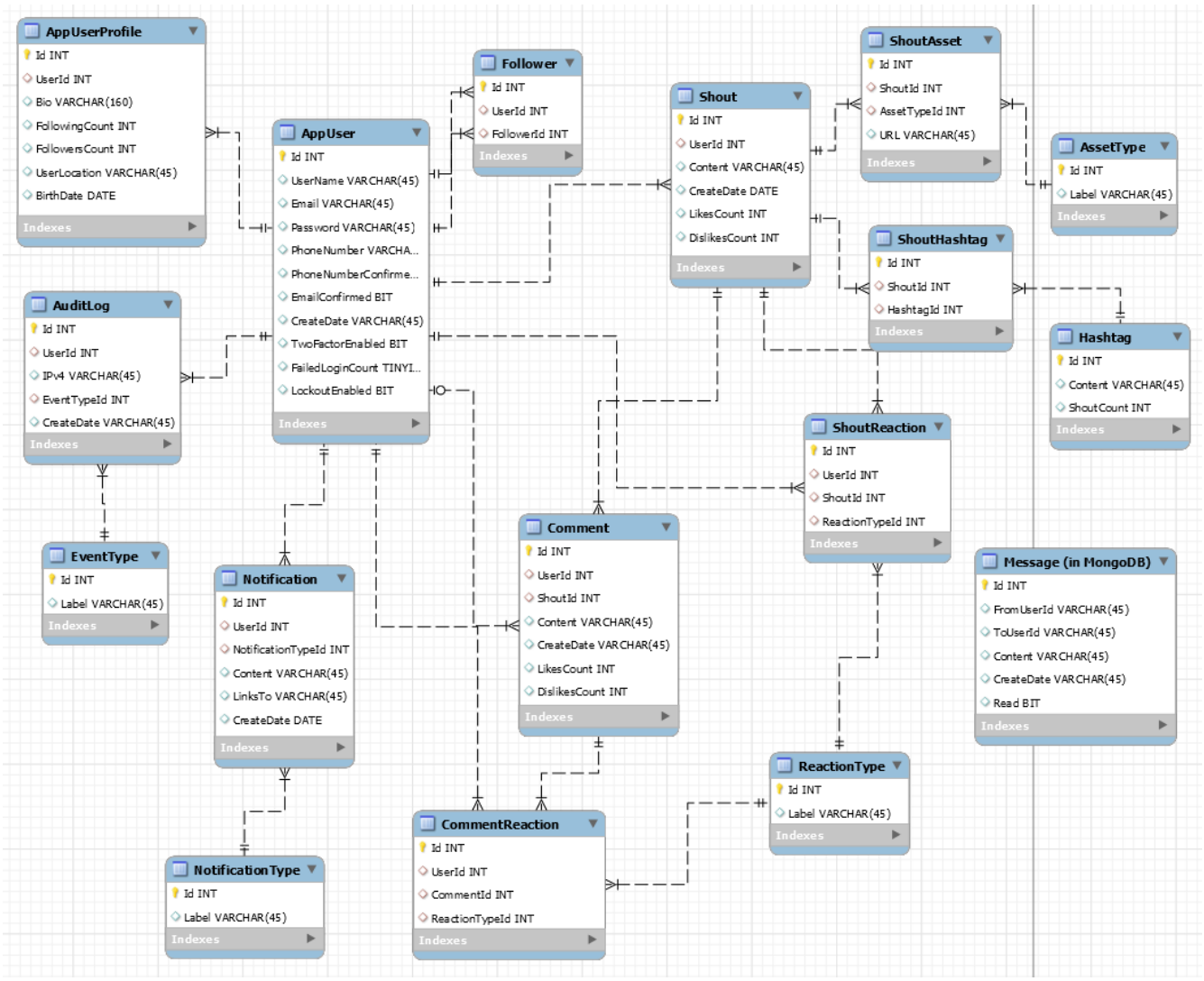
The following diagram describes the database schema.

The developer should use it only as a visual representation and he must use the code present in the project "*Amplifir\Amplifir.Infrastructure.DatabaseCode*" as the true and complete representation of the DB, with its definitive data types, checks, indexes, etc.

When a change is made to the database schema, it must be reflected on the *Infrastructure.DatabaseCode* project's code before deployment. Only after that, the diagram and this document should be updated.

The code must be written by hand and not auto-generated.

The diagram is made using MySQL Workbench diagram tool.



API Documentation

The API documentation is generated through NSwag, using the OpenAPI definition and the proxy API code through Nswag Studio.

It is available through “https://<Amplifir-link>/swagger”.