

# *Chirp!* Project Report

ITU BDSA 2024 Group 6

Benjamin Ormstrup beor@itu.dk

Marcus Frandsen <megf@itu.dk

Victor de Roepstorff vicd@itu.dk

Valdemar Mulbjerg vamu@itu.dk

Christian Jörgensen chpj@itu.dk

# 1 Design and Architecture of Chirp!

## 1.1 Domain model

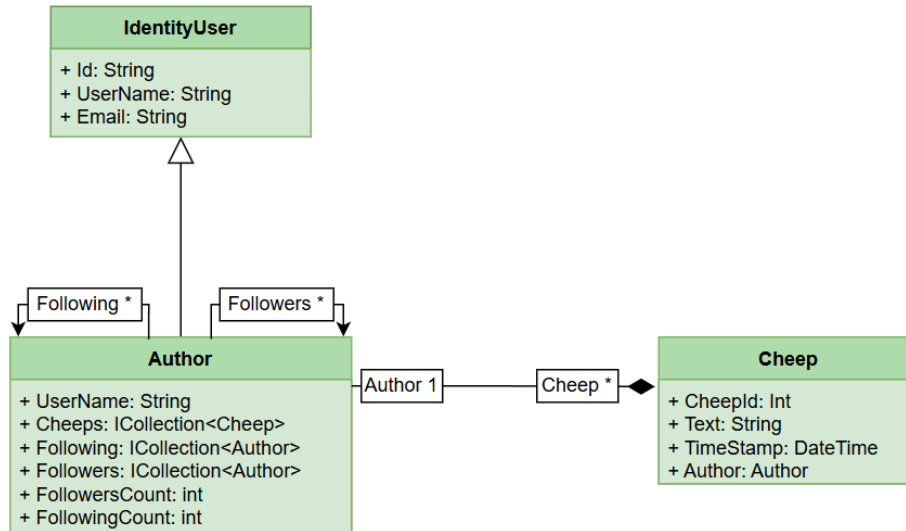


Figure 1: *Illustration of the Chirp! domain model*

The UML class diagram above provides an overview of the core domain model for our *Chirp!* application, highlighting the primary entities, their attributes, and the relationships between them. A key aspect of the diagram is the cardinality between the entities, which defines how objects in one class relate to objects in another.

## 1.2 Architecture — In the small

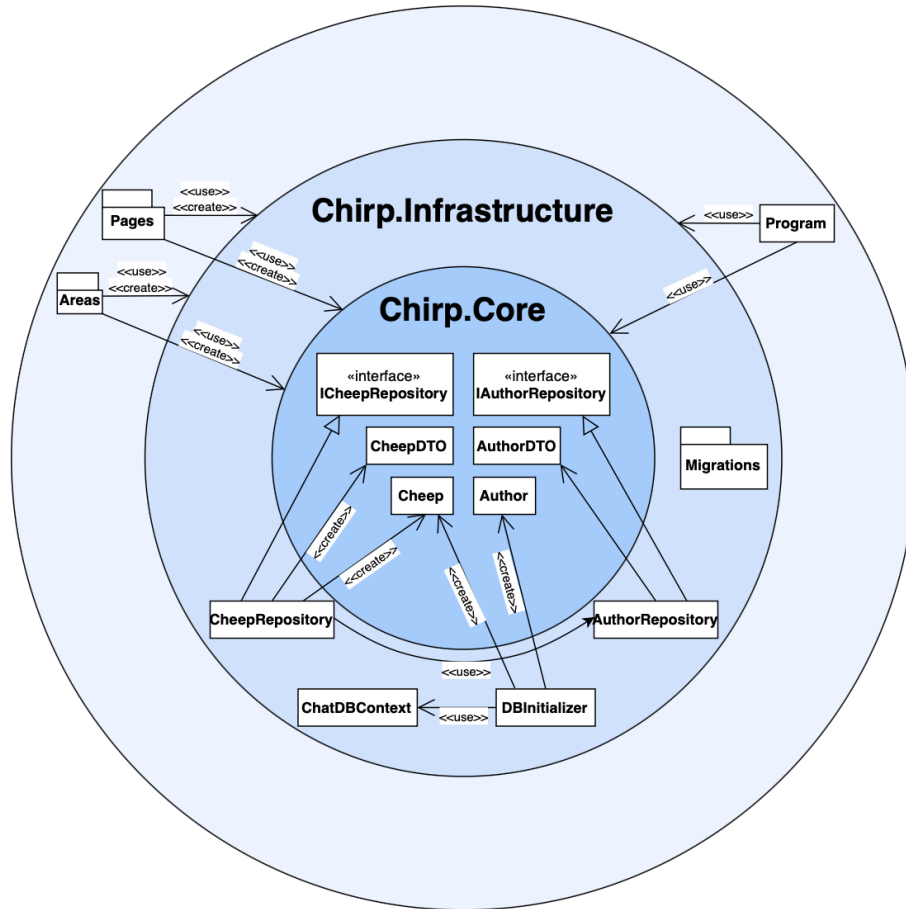


Figure 2: *Illustration of the Chirp! architecture*

The union architecture diagram above visually represents the layered structure of the Chirp! application. The diagram consists of three circles with a different shade of blue, each symbolizing one of the core architectural layers: Core, Infrastructure, and Web. The components within each circle represent the key parts or responsibilities of that layer. The arrows throughout the diagram represent the unidirectional dependency flow of the application, where each layer depends only on the layers inside of itself:

- The Core layer is independent and does not depend on any outer layer.
- The Infrastructure layer relies on the Core while still remaining separate from the Web layer.

- The Web layer depends on both the Infrastructure and Core layers to deliver functionality to the user.

This layered structure ensures separation of concerns, making the program easily maintainable, testable, and scalable. Each layer can be adjusted, without having a direct impact on the logic and functionality of the layers above it.

### 1.3 Architecture of deployed application

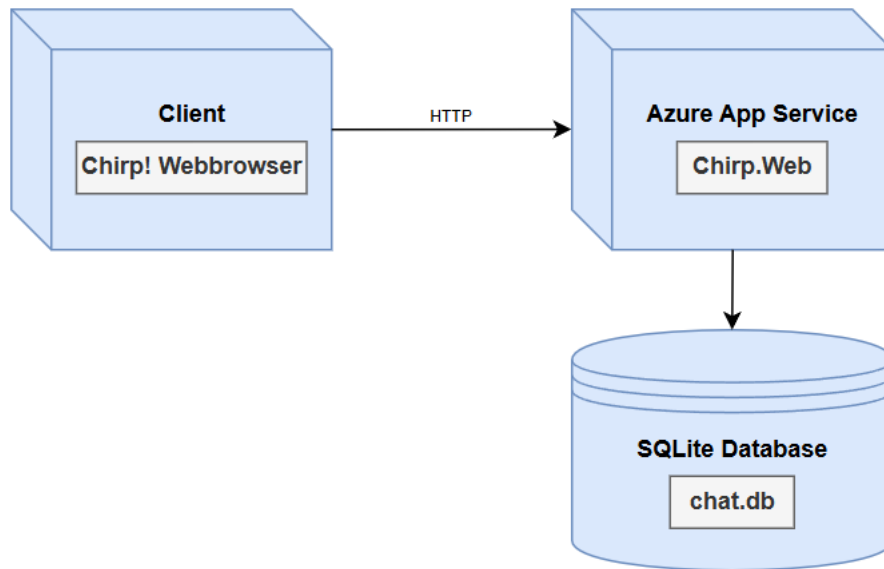


Figure 3: *Application Architecture diagram of the deployed application*

Above is a diagram that illustrates the architecture of our deployed Chirp! application which focuses on the client-server relation.

1.4 User activities

1.5 Sequence of functionality/calls through Chirp!

## 2 Process

2.1 Build, test, release, and deployment

2.2 Team work

2.3 How to make Chirp! work locally

2.4 How to run test suite locally

## 3 Ethics

3.1 License

3.2 LLMs, ChatGPT, CoPilot, and others