Chirp! Project Report

ITU BDSA 2023 Group 5

Jakob Arnfred arni@itu.dk Johan Brandi johbr@itu.dk
Niklas Zeeberg Hessner Christensen nizc@itu.dk
Olivier-Baptiste Hansen oliha@itu.dk
Philip Guozhi Han Pedersen phgp@itu.dk

1 Design and Architecture of *Chirp!*

1.1 Domain model

Here comes a description of our domain model.

1.2 Architecture — In the small

1.3 Architecture of deployed application

1.4 User activities

1.5 Sequence of functionality/calls trough *Chirp!*

2 Process

2.1 Build, test, release, and deployment

The following UML activity diagrams illustrate the GitHub actions workflows that are run when different criteria are met. This will be briefly described under the respective diagrams.

![UML activity diagram of the build and test workflow.][ht](images/Build_test_release_and_deployment/build_a

This workflow is run upon every push to main and pull request to main. It builds and tests to application in order to keep main void of faulty code (as **safety net**).

Illustration of the *Chirp!* data model as UML class diagram.

Figure 1: Illustration of the *Chirp!* data model as UML class diagram.

![UML activity diagram of the deployment workflow.][ht](images/Build_test_release_and_deployment/deployme

This workflow is run upon every push to main. Note the redundant “build” step. We do not need this since the “publish” step already builds the application. This redundancy was not noticed during development and has not been removed due to time constraints.

![UML activity diagram of the release razor workflow.][ht](images/Build_test_release_and_deployment/release_

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