

Assessment for vkhorkov's team

Krb5 repository url

https://:@gitlab.ucode.world:8443/ucode/uchat/vkhorkov.git



Krb5 ▾

Media



uchat

General

1. The goal is to share the experience with **assessor** and **defender** about this challenge.
2. Evaluate thoroughly the defending team's code in the specified repository. You are responsible for an objective grade of the team's work and knowledge.
3. Clone the repository.
4. Verify the correctness of the submitted solution according to the **Auditor** rules. If at least one rule has been violated, indicate it.
5. Files must be in the corresponding directories with names as specified in the story. If this isn't true, indicate it.
6. Be rigorous and honest, use the power of p2p and your brain.
7. If you have a disagreement, refer to the p2p and defenses documents.
8. Compile C-files with clang compiler and use these flags: `clang -std=c11 -Wall -Wextra -Werror -Wpedantic`. If the project does not compile, invalidate it.
9. Correct only the files in the cloned repository. If the story has a SUBMIT section, only the specified files should be present. If the story describes a product layout, the product must follow it. Indicate if the program does not meet at least one of these two requirements.
10. There can be several different types of questions in the protocol. Answer them according to the rules below:
 - **Binary a.k.a. true-false** - mark as **True** only if everything works perfectly according to the question, or leave **False** if something fails for at least one case
 - **Range 0-10** - add points strictly according to the instructions in the question
 - **Label** - select one specific label according to the case detected during the entire assessment
 - **Checkbox** - select all options which are appropriate to the question
 - **Comment** - leave a descriptive and understandable comment to the question
11. Carry out the evaluation only in the presence of all members of the defense. Postpone assessment until all defense participants can come together.
12. Exchange knowledge during the assessment.

Assign a label below

Select the first option that came up.

Help the defending team to understand their mistakes, discuss the challenge in detail, and exchange knowledge.

Repository

► The repository is empty. You have nothing to evaluate, select this item.

Auditor

► There is at least one mistake according to the Auditor.

Compiling

► The code in the repository does not compile.

Crashing

► The challenge solution crashed (segmentation fault, abort, bus error, uncaught exception, etc.) at some point during the correction.

Irrelevant files

► The solution doesn't meet the story requirements about files or solution layout. Select this item if there are any files that are not part of the solution, or the directory layout doesn't match the one described in the story.

Cheat

► The defender used any form of cheating. Cheating involves actual, intended, or attempted deception and/or dishonest actions in relation to any academic work of ucode. Everything that

is submitted must be understandable, justified, and explained.

OK

► If none of the issues above were found.

Act

The following set of questions tests if the project complies with the **Act** of the story.

Carry out the assessment honestly and in accordance with the challenge.

Everything must work correctly, and without runtime, compilation, or logical errors.

An honest assessment is much more valuable than overestimated or underestimated marks.

First execution

Do a basic test of the solution.

Make sure that the program consists of two parts, the **client** and the **server**.

The program launches correctly.

FALSE

TRUE

Error management

All kinds of user behavior work through the **client** without errors.

The **server** works without internal server errors.

In case of any error, the program displays clear information about it.

FALSE

TRUE

Authentication & Authorization

Start the server and the client on the first computer.

Try to authenticate.

Create an account and log in. Log out.

The authentication and authorization processes work as expected.

FALSE

TRUE

Send message

Log in into another computer in the cluster and install the **client** app.

The client connects to the server running on the first computer.

Create one more user and send a message to the first user.

Test the message exchange from both users.

Message sending works correctly.

FALSE

TRUE

Messenger

The messaging app enables communication among users.

There are such features in the product:

- ☐ Edit messages ▶ Edit already sent messages. The messages are changed for all users.
- ☐ Delete messages ▶ Delete sent messages. The messages are deleted for all users.
- ☒ Unicode ▶ A very important feature for the whole world is sending messages in different languages. This feature is implemented and works correctly.
- ☐ Forward messages ▶ Forward the message to another user or group.
- ☐ Reply to message ▶ Comment/reply to specific posts. The users easily understand which message was replied to.
- ☐ None ▶ None of the options apply.

Attachments

Modern messaging apps have different types of data, such as files/stickers, etc.

- ☐ Images ▶ Send and preview pictures/photos.
- ☐ Files ▶ Send files of various file types, such as `.zip`, `.txt`, etc.
- ☐ GIFs ▶ Send and preview GIFs and other types of animations.
- ☐ Stickers ▶ Send an illustration of a character that represents an emotion or action. It must be associated with a sticker pack.
- ☐ Emoji ▶ Send emoticons along with a regular text message.
- ☐ None ▶ None of the options apply.

Notifications

The messaging app provides different types of notifications for events.

There are such features in the product:

- ☐ Visual ▶ Some kind of a pop-up notifying about an event.
- ☐ Email ▶ Email notifications, e.g. an aggregated summary during inactivity time or any other events.
- ☐ Sound ▶ An audio signal indicating about new events.
- ☐ Notification settings ▶ Configure types, muting, visual representation and other settings of notifications.
- ☐ None ▶ None of the options apply.

Data

Database deployment is performed with a Makefile.

The database is used as storage for all necessary data.

FALSE TRUE

Contacts

There is an ability to add contacts to various groups, e.g. **Friends** for quick access to necessary users.

Try to add and remove user contact to/from **Friends**, or another custom group.

Everything works correctly.

FALSE TRUE

Security

The users' confidential information is protected and won't fall into the hands of abusers.

The program supports encryption of chats and personal data.

The users are provided with a reasonable level of security.

Search

The messaging app has search by different items, such as messages, contacts.

It is a special area for searching information.

There are such features in the product:

- ☒ Users ▶ Find people according to the contact data.
- ☒ Messages ▶ Find messages in the chat history, it's essential to find all occurrences of the query.
- ☐ None ▶ None of the options apply.

Groups/channels

There are many advanced features that the students could have implemented.

There are such features in the product:

- ☐ Groups/channels ▶ Create and subscribe for multiuser groups/channels.
- ☐ Permissions ▶ Distribute the access rights for group/channel members by some rules.
- ☐ Ban ▶ An ability to blacklist or restrict permissions of a group/channel's members.
- ☐ None ▶ None of the options apply.

Formatting

The program supports custom syntax highlight or text formatting.

Add 2 points for each formatting option that works correctly (do not forget to test them completely).

For example, bold, italic, text color, text size, etc.

User profile

The users manage profile information.

- ☐ Profile ▶ Display and edit user information, e.g. name, surname, some status or maybe nickname.
- ☐ Edit profile ▶ Edit information which was entered before.
- ☐ Blacklist ▶ Block certain contacts to communicate with the user in any way.
- ☐ Delete account ▶ Delete personal account.
- ☐ Multi-users ▶ Multiple accounts on one device. Try to add one more account and test it.
- ☐ None ▶ None of the options apply.

Audio

Audio content is recorded and played in the app environment.

Add 4 points if the program allows audio recordings in real-time and sending them afterwards.

Add 3 points if the program supports music playback in at least one standard audio format (e.g. mp3).

Add 3 points if the program implements music player features, for example, play, stop, pause, rewind, fast forward, etc.

Video

The video content is recorded and played in the app environment.

Add 4 points if the program allows video recordings in real-time and sending them afterwards.

Add 3 points if the program supports video playback in at least one standard video format (e.g. mp4).

Add 3 points if the program implements video player features, for example, play, stop, pause, rewind, fast forward, etc.

Themes

The program supports different themes that change the appearance of the app.

Themes are customizable through the settings.

Themes settings are saved even after reopening the window.

The theme does not affect any functionality of the app and is pleasant to the eye.

Do not validate this question if the theme is not user-friendly and has toxic color palettes.

☐ FALSE☐ TRUE

UX/UI

The app has a user-friendly UI.

All UI elements have smart placement, are accessible and correspond to the whole project design.

Also, all the features have a simple way to use, with the most important features being most accessible.

All the input and output information is readable and easy to interact with.

Rate on the scale of 0 (not implemented) to 10 (beautiful and intuitive design).

Multi-language

The app supports changing the app language in the settings.

The language is saved even after restarting the app.

All app text is changed to the selected language.

Try to do it several times.

☐ FALSE☐ TRUE

Unit testing

Unit testing is a way to check the correctness of the app features in automatic mode.

Tests are important part of product development.

The program implements unit tests which are related to the main features of app.

Try to run it and check the results.

☐ FALSE☐ TRUE

More cool features

Add 1 point for each additional creative feature that works and is useful.

Adding a feature just for the sake of adding one is not a product-oriented approach.

Test each feature thoroughly.

If you are not sure a feature is useful, then ask other students.

Big idea

A great product idea is 50% of success.

Every product has its own big idea.

It explains the causes of product development.

The idea of the messaging app is excellent.

Do not validate this question if the messenger was created just to complete the task.

Reflection

Evaluate how well the defending team members understand the learning process they went through, and their progress.

Talk to the team, discuss its answers in the reflection protocol.

Evaluation

How detailed, meaningful, and clear are the responses?

Maximum grade if the answers are well-detailed, and they clearly reflect the essence of the challenge.

Minimum grade for short and/or poorly written responses.

It's okay to score low marks, the reflection process can be difficult.

[Link to Reflection](#)

Document

Check if the team has documented the challenge phases.

Documentation

The team completed documentation of the challenge.

All phases have been documented from challenge to solution.

For this, the team used the appropriate tools (README, Google Tools, Dropbox Paper, Git Wiki, Haroopad, Canva, etc.).

Code commenting is present in the implementation.

So, how do you like the documentation for this project?

Share

Check if the team has shared the information about this challenge.

Publishing

The team shared its solution with the world.
They have shared their work on GitHub/GitLab/BitBucket or something similar.
They wrote an article, or a post on social media about the challenge.
If at least one of these two statements is true, mark as true.

FALSE

TRUE

Feedback

Your feedback on the evaluation.

Comment

Leave a comment on this evaluation.

Comments *

Nice one

Finish Assessment