Qualifying Scala project

Telegram Bot

In this project we suggest you to implement a telegram bot with custom functionality. This project will consist of two parts as follows:

1. Telegram Requests Handling

The basic part which should be present in all projects of this type. You will be asked to implement a backend service to handle any Telegram user's input to the bot. To complete this part of the project you should present a telegram alias of your bot and any student or TA should be able to initiate communication with the bot by typing the /info command. The bot should respond with a detailed explanation of it's functionality, list of possible commands and the link to the github repository.

To do all the above you will have to:

- create a special bot token via Telegram's Bot-Father(@BotFather) - a special bot to create new bots.
- create a Scala REST server to handle user input to the created token.

For the REST server we recommend to use one of the following:

- Scalatra
- Play Framework
- Akka HTTP

2. Functionality

Keep in mind that these are just ideas and not a detailed documentation. Specify your functionality to the TAs before commencing work on this part of the project.

By this time you will have to come up with an idea what to write in your /info response. Let's try to make your bot a bit useful. We propose you with a few ideas, but you can come up with your own.

- Reddit Crawler
    - Create another service that will parse a selected subreddit.
    - user of your bot can initiate the following command to the bot: /stalk *SUBREDDIT_NAME* {*WHEN*}, where the {} part is optional. The bot should respond with a special ID for this stalkering, so the user can cancel it at any time. This command will be handled by your telegram-bot backend and will make a request to your Reddit Crawler service, which in return will download the *SUBREDDIT_NAME* front page, parse top-5 topics names and links, store them until 9:00 AM (or *WHEN*, if given) and send them to the bot-backend at this time each day, until canceled with a special /stop *ID* command from the user. This cycle should then continue until canceled.
- Week Scheduler
    - Create a service for scheduling user's week from the bot.
    - As a user I want to initiate commands to the bot like /keepInMind *DAY_OF_THE_WEEK* *TIME* *ACTIVITY_NAME* *DURATION*. This will create a record for the current user and will notify him a day before in the evening of the upcoming agenda and on the specified morning, so he wont forget it.
    - The bot should pay close attention to the duration of the user activities so that they would not interfere.
- Meme aggregator
    - Use imgur.com API (https://api.imgur.com/) to build your own MEME channel!
- Any other idea you may come up with. (aka Contact your TA for futher instructions)

The grading will go like this:

- for C grade, you will have to demonstrate a functioning (at least to some degree) bot from one of the ideas above. Once you achieve this functionality, please, consider adding features for B grade!
- for B grade, you will implement features, that will be given to you by your TA, unique for each team. As soon as you're finished with adding new stuff, please, consider improvements for A grade.
- for A grade, you will just have to make a detailed README page for your git repository, provide usage examples (possibly with screenshots), write all necessary tests and deploy your services on some hosting, so that everyone could use your creation at anytime :) .