

КАФЕДРА _____

ОТЧЕТ ПО УЧЕБНОМУ ПРАКТИКУМУ

подпись, дата	название.
---------------	-----------

подпись, дата

фамилия, и.о.

Черненко В.

подпись, дата фамилия, и.о.

Оленев А. О.

подпись, дата

фамилия, и.о.

2020 г.

Оглавление

Введение.....	3
Аналитический раздел.....	4
Конструкторский раздел.....	5
Технологический раздел.....	7
Заключение.....	8
Литература.....	9

Введение

Перед командой была поставлена задача в отведённый срок реализовать проект. Целью проекта было создание Telegram bot "MergeBot", предназначенного для упрощения процесса разработки программного обеспечения путем интеграции хостинга репозитория GitLab с популярным мессенджером Telegram. Он оповещает пользователя о новых совершенных merge request.

Роли, закрепленные за участниками команды.

Климов Илья – тимлид, разработчик (2-3 стр)

Богатырев Иван – техлид, разработчик (7 стр)

Трунов Андрей – разработчик (6 стр)

Сысоева Виктория – технический писатель, ответственная за базу данных (4 – 5 стр)

Межеровский Александр – разработчик (8 – 9 стр)

Аналитический раздел

Обзор существующих решений.

Во время написания кода не было необычных технических решений. При начале работы с ботом необходимо написать вам доступно две команды /start и /problem, в остальных случаях вы увидите сообщение о не существовании такой команды.

Для начала работы введите команду /start.

Бот поприветствует пользователя в зависимости от времени суток, и, если пользователь уже есть в базе данных выведет сообщение с токенами для выбора и также предложит ввести очередной токен по желанию. Если же пользователя нет в базе, то запишет его Telegram id и предложит ввести токен, который необходим для того, чтобы получить доступ к репозиторию. Далее бот предложит авторизоваться на GitLab, после этого пользователь сможет получать уведомления о merge request, его закрытие и новых коммитах.

Еще одной командой является запрос /problem.

Она необходимо для обратной связи. Если вы нашли ошибку в нашем боте, напишите нам об этом.

Еще одной командой является запрос /help.

Она дает инструкцию пользователю, которая поможет ему запустить бота.

Анализ достоинств и недостатков.

Достоинством можно считать простоту реализации, при этом это решение отвечает всем поставленным задачам. Также к достоинствам можно отнести user friendly interface.

Недостатком можно считать то, что вероятная заторможенная работа бота при одновременным использование бота большим количеством пользователей.

Обоснование необходимости нового решения.

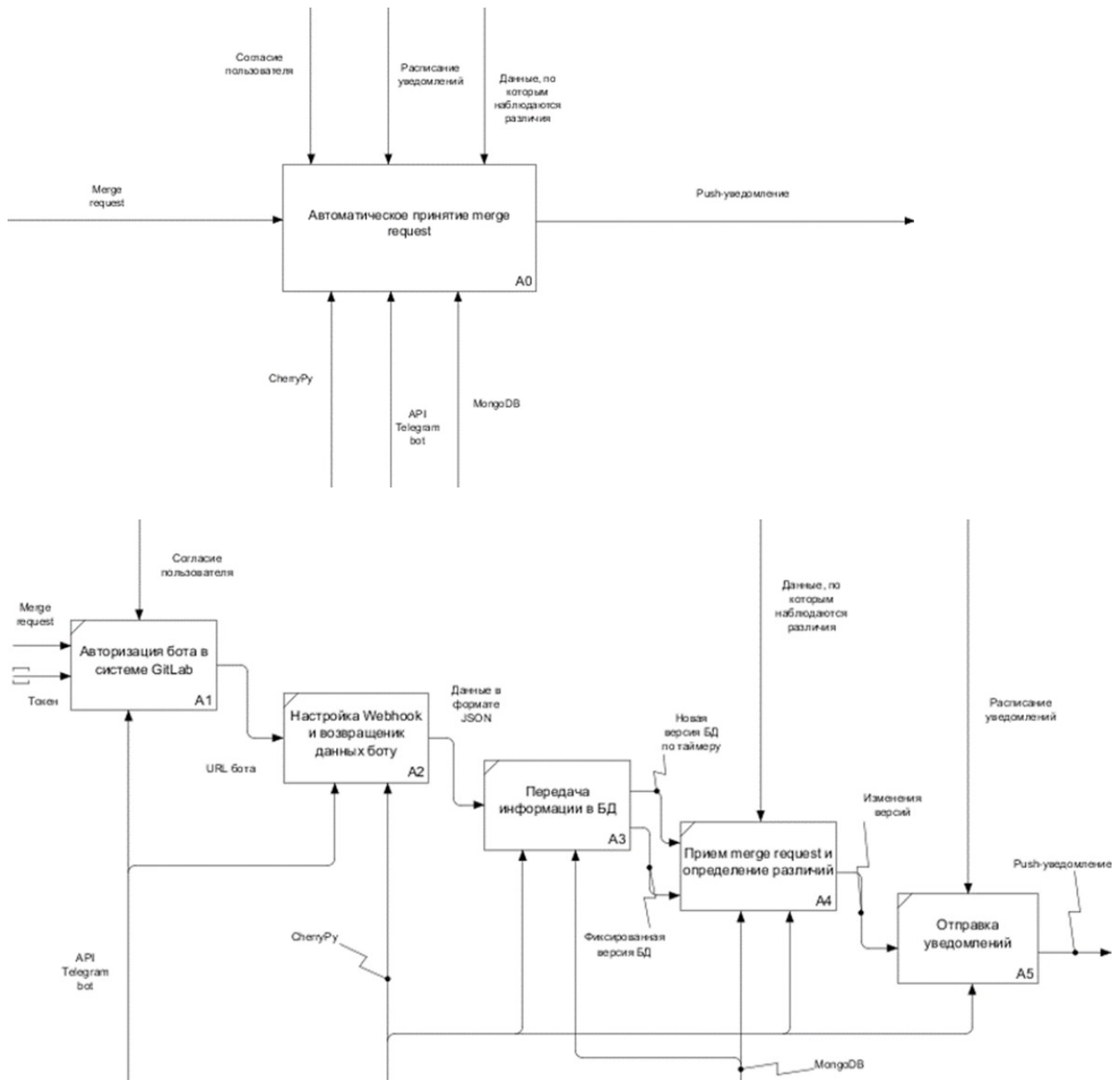
Мы считаем, что наш Telegram bot должен облегчить процесс разработки ПО, улучшить взаимодействие между разработчиками, а также упростить и ускорить процесс принятия Merge Request.

Постановка задачи.

Создать Telegram, способного уведомить пользователя о новом поступившем Merge Request и возможностью просмотра внесенных изменений.

Конструкторский раздел

Декомпозиция задачи.



Разработка структуры создаваемого продукта.

Платформой для реализации поставленной задачи был выбран мессенджер Telegram. Telegram позволяет создавать чат-ботов, которые можно наделять различным функционалом. В ответ на команды пользователя выдает соответствующие сообщения и кнопки.

Описание назначений, требований к выделенным компонентам и их интерфейса.

Кнопка «Ввод TOKEN» - позволяет пользователю добавить токен.

Кнопка «Выбор TOKEN» - позволяет пользователю выбрать токен из ранее введенных.

Кнопка «Удаление TOKEN» - позволяет пользователю удалить ранее введенный токен.

Кнопка «Как получить TOKEN» - ссылка, помогающая понять пользователю откуда взять токен.

Разработка алгоритмов и структур данных для выделенных компонентов.

Для работы бота была создана база данных, в которой хранится информация о пользователях, а именно idTelegram, idGitLab и токены, введенные им.

Один из использовавшихся алгоритмов — поиск по полю базы данных id Telegram словаря токенов, для того чтобы записать туда новый или выдать все используемые пользователю.

Среди структур данных можно отметить работу с словарем, который использовался для хранения токенов, и файлом, в котором прописана информация, дающая доступ боту к базе данных.

Проектирование пользовательского интерфейса.

Не разрабатывали собственный пользовательский интерфейс. В нашем случае интерфейс — диалоговое окно с ботом в Telegram.

Описание способов тестирования выделенных компонентов и программного продукта в целом.

Тестирование происходило при непосредственном запуске бота на различных этапах разработки, путём взаимодействия с ним. После обращения к боту проверялось отвечает ли он соответствующими сообщениями на команды, а после использования кнопок, просматривалась база данных и проверялось заносятся ли соответствующие данные.

Технологический раздел

Выбор и обоснование технических средств (язык, БД, библиотеки).

Язык программирования Python 3

При разработке кода: библиотеки telebot, pymongo, cherrypy, python-gitlab

База данных: MongoDB

API Telegram бота – pyTelegramBotAPI

API для работы с GitLab – python-gitlab

Выбор и обоснование модели разработки.

Была выбрана Waterfall Model (каскадная модель или «водопад») модель. Эта модель подходила для создания в сжатые сроки.

Реализация программного продукта.

Изначально был сделан бот, который имел доступ к базе данных, далее был настроен сервер для возможности развертывания бота на его основе. Далее была разработана система записи данных пользователя в базу данных. После того, как механизм взаимодействия пользователя с данными был закончен и протестирован, была начата работа над интеграцией авторизации на GitLab и возвращение сообщений о merge request.

Реализация тестирования.

Основным видом тестирования стало модульное тестирование.

Так как присутствовало исправление ошибок, как и добавление новых возможностей, сильно изменяющих программу, впоследствии проводилось регрессионное тестирование. Ручное тестирование многократным запуском помогало выявить ошибки.

Также происходило тестирование интерфейса. Тестирование различных вариантов оформлений позволяло выбирать наиболее понятные пользователю.

Развертывание разработанного программного продукта.

Установки в привычном её понимании не требуется, необходимо установить на свое устройство мессенджер Telegram и найти в поиске бота "MergeBot" после чего написать ему /start и начать работать с ним.

Заключение

Все задачи, поставленные перед нами, были успешно выполнены. Мы получили готовый продукт, который можно продемонстрировать. В дальнейшей разработке возможно добавление поддержки всех типов событий (commit, push, etc.) и интеграция с другими аналогичными сервисами (например, GitHub или BitBucket).

Литература

1. Изучаем Python. Программирование игр, визуализация данных, веб-приложения
Автор: Эрик Мэтис
 2. Пол Бэрри. Изучаем программирование на Python
 3. JavaScript and JSON Essentials Author: Sai Srinivas Sriparasa
 4. Документация на библиотеку pyTelegramBotAPI
<https://pypi.org/project/pyTelegramBotAPI/>
 5. Документация на CherryPy <https://docs.cherrypy.org/en/latest/>
 6. Документация на PyMongo <https://pymongo.readthedocs.io/en/stable/tutorial.html>
- .