План проекта по созданию Telegram-Bot аптеки.

1. Цель проекта

Создать Telegram-Bot, который будет упрощать работу пользователям с товарами ближайших аптечных пунктов. Результатом данного проекта должен быть прототип Telegream-Bot «CureMe».

1. Роли
2. Анна Ромме – программист разработчик. Отвечает за Core программного продукта и внедрение дополнительных вспомогательных функций, необходимых для полноценного функционирования проекта.
3. Иванова Татьяна – программист разработчик. Отвечает за Core программного продукта, основные структуры и типы данных проекта.
4. Барабанова Ирина – программист-менеджер. Отвечает за документацию к проекту, коммуникацию и внедрению вспомогательных функций.
5. Одилов Азимбой – программист тестировщик, администратор БД. Отвечает за обеспечение качества работы продукта и тестовых данных.
6. План и объем выполняемых работ.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MegaStage | Stage | Result | Initial date | Finite date |
| Анализ | **Сбор требований** | Документ с Требованиями к проекту | 09.03.2020 | 22.03.2020 |
| Проектирование | **Разработка архитектуры** | UML-use-case диаграмма | 09.03.2020 | 22.03.2020 |
| **Разработка календарного плана** | Диаграмма Ганта | 23.03.2020 | 05.04.2020 |
| **Составление плана проекта** | Документ с планом проекта | 23.03.2020 | 05.04.2020 |
| Разработка и тестирование | **Реализация кода** | Прототип бота | 06.04.2020 | 10.05.2020 |
| Профилактические работы | **Реализация кода, подготовка документов** | Отчет о тестировании,  Документация готового продукта | 11.05.2020 | 25.05.2020 |
| Внедрение |  | Отчетная презентация | 25.05.2020 | 07.06.2020 |

1. Инструменты, используемые в проектировании и разработке продукта.
   1. Trello – менеджер управления проектами с использованием Kanban досок. Использование расширения Elegant для Google Chrome для отображения диаграмм Ганта.
   2. Использование платформы TravisCI для непрерывной интеграции проекта.
   3. Использование IDEA PyCharm для разработки проекта.
2. Методология разработки проекта.
   1. При планировании проекта применены 2-ух недельные спринты по принципу Agile. Гибкая методология применена в силу небольшого размера проекта и небольшого размера команды.
   2. Задачи ставятся исходя из собранных требований и поправок в ходе разработки.
   3. Количество задач на спринт для каждой роли назначается исходя из имеющихся человеко-часов. В среднем на спринт ставится 1-2 задачи на человека при наличии 15 часов/нед.
   4. Прототип проекта планируется завершить к концу мая 2020 года. План основных этапов разработки ПО представлен на Рисунке 1.
3. Основные задачи разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Description | Use-Case | Developer |
| Реализация команды /start | Реализовать функцию обработки команды /start   1. При введении команды /start должен появляться сообщение с дальнейшими рекомендациями и возможными командами. | При инициализации работы с ботом пользователь должен понимать, что ему дальше можно сделать. | Romme Anna |
| Реализация команды /help | Реализовать функцию обработки команды /help   1. Ответом на команду должно быть справочное сообщение | При вызове этой команды пользователь должен получать полную справку со всеми доступными командами | Azimboy Odilov |
| Реализация команды /register | Реализовать функцию обработки команды /register | После введения команды /register должен инициироваться процесс регистрации | Ivanova Tatiana |
| Реализация команды /updatelocation | Реализовать функцию обработки команды /updatelocation   1. Ответом на команду должен быть возврат к процессу инициализации запроса геоданных 2. Должен предоставляться список ранее регистрируемых адресов для данного пользователя, если таковые имеются | После введения команды пользователю должна предоставляться инструкция по дальнейшим действиям по смене локации | Ivanova Tatiana |
| Функция подключения MainDB. | 1. Создать тестовую базу данных для регистрации пользователя с полями: ID(key), Город, Улица, Дом 2. Реализовать функцию инициализации подключения к базе данных. | Нет прямого применения пользователем | Azimboy Odilov |
| Функция подключения CacheDB | В CacheDB хранятся все ранее используемые адреса для данного пользователя.   1. После обновления полей в MainDB для конкретного пользователя должна производиться запись полного Адреса (цельной строкой) в CacheDB |  | Azimboy Odilov |
| Функция обращения к API  (Core Task) | 1. Инициализация обращения к API 2. Функция поиска необходимого товара 3. Сборка результирующего выходного сообщения, включающего: цену, название аптеки, адрес, количество штук в наличии, необходим ли рецепт. | После введения названия товара пользователь должен получать сводную информацию по найденному лекарству. | Ivanova Tatiana |
| Функция сборки результата запроса | 1. Дизайн результата запроса 2. Функция вывода собранного результата пользователю | Пользователь должен получить таблицу с возможными вариантами, где можно купить товар. | Barabanova Irina |
| Интерфейс корзины | 1. Дизайн корзины 2. Функция добавления в корзину 3. Функция очистки из корзины | Пользователь должен видеть все выбранные товары в корзине | Barabanova Irina |

1. На блок-схеме представлена логика работы Telegram-Bot

A picture containing text

Description automatically generated

Result output example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | PharmacyName | PharmacyAddres | Available | Recipe |
| Arbidol | Ozerki | пр. Елизарова, 12, Санкт-Петербург, 192029 | 3 | No |

Result Bag example:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | PharmacyName | PharmacyAddres | Number |
| Arbidol | Ozerki | пр. Елизарова, 12, Санкт-Петербург, 192029 | 1 |

1. Основная архитектура приложения представлена на диаграмме классов. A close up of text on a white background

   Description automatically generated

На рисунке1 описаны основные этапы жизненного цикла проекта.

A screenshot of a computer

Description automatically generated

Рисунок 1

Детализированный план этапа Разработки с назначенными датами и ролями представлен на Рисунке2.

A screenshot of a social media post

Description automatically generated

Рисунок 2