Министерство науки и высшего образования РФ

КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ

МИППС

Контрольная работа №1 вариант 1

По методологии проектирования, разработке и внедрению информационных систем

Студент 19-ЗКМ-067 2 курса, шифр 09.04.04

Направление Программная инженерия

Фамилия Ханабеев

Имя Антон Отчество Васильевич

Дата поступления работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка

Рецензент Попова Ольга Борисовна

« » Подпись

СОДЕРЖАНИЕ

[1 Методология функционального моделирования 3](#_Toc62257058)

[2 Методология объектно-ориентированного моделирования 7](#_Toc62257059)

[3 Методология управления проектами 10](#_Toc62257060)

[4 Использование систем контроля версий исходного кода программ 14](#_Toc62257061)

# 1 Методология функционального моделирования

При разработке информационных систем на этапах кодирования и тестирования обнаруживается большое количество ошибок, исправление которых может повлечь за собой кардинальное изменение всей разрабатываемой системы. Возможные ошибки учитываются при моделировании. Моделирование позволяет «увидеть» систему в процессе разработки и создать предпосылки для анализа поведения системы в зависимости от начальных условий.

Для правильного распределения процессов протекающих в моделированной системе необходимо создать структуру, т.е. упорядочить процессы. Моделирование работы информационной системы особенно важно на первых этапах её создания.

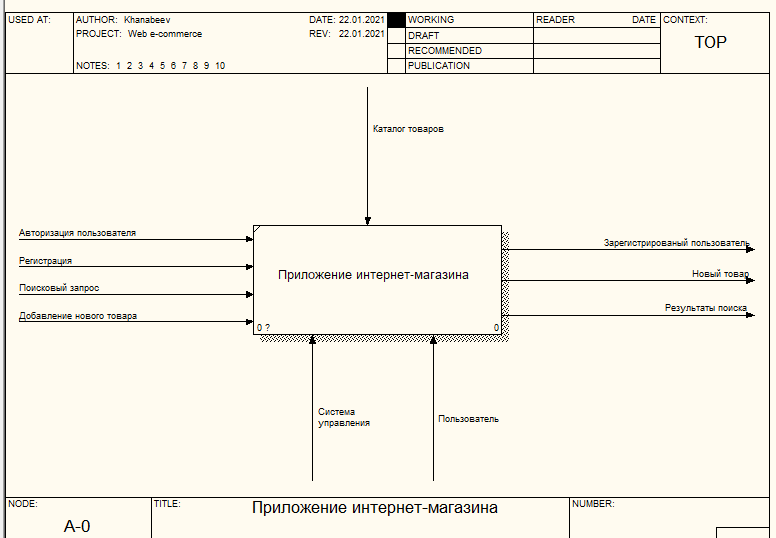


Рисунок 1 - Контекстная модель

Декомпозируем контекстную модель на несколько этапов.

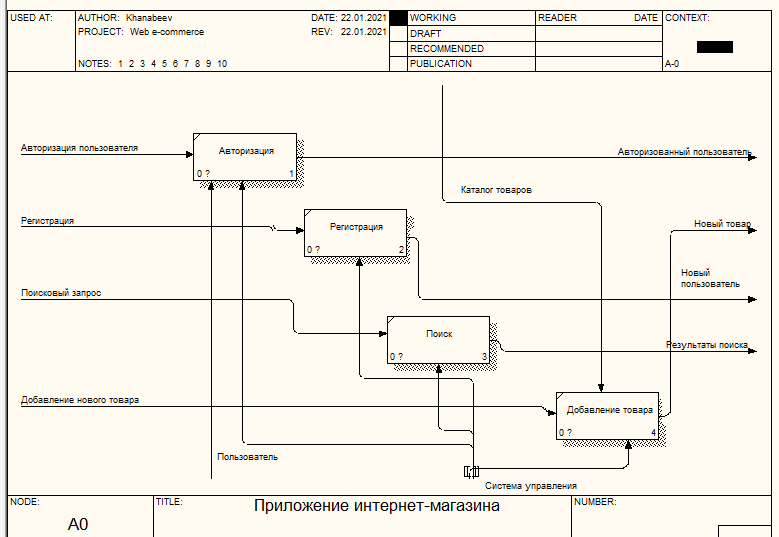


Рисунок 2 - Декомпозиция контекстной модели, диаграмма 1-го уровня

Декомпозируем блок Автризация. Данная диаграмма имеет тип IDEF3

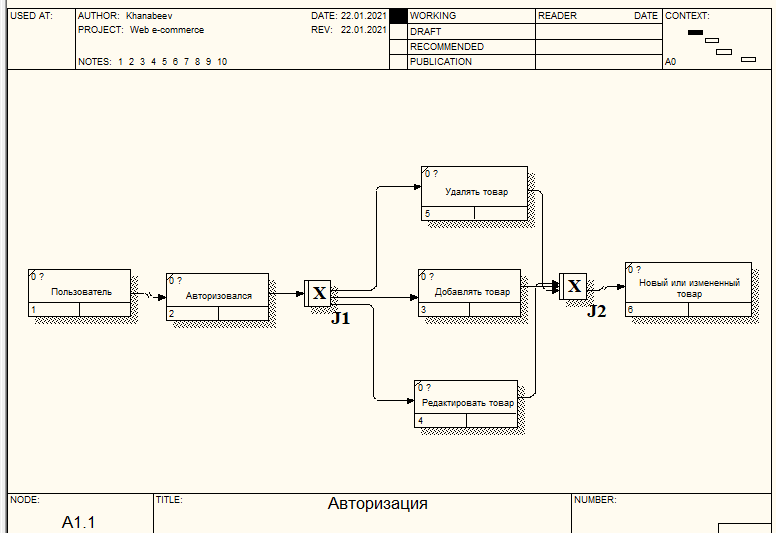


Рисунок 3 - Декомпозиция функционального блока аворизация, модель IDEF3 первого уровня

Декомпозируем блок «Редактирование товара»

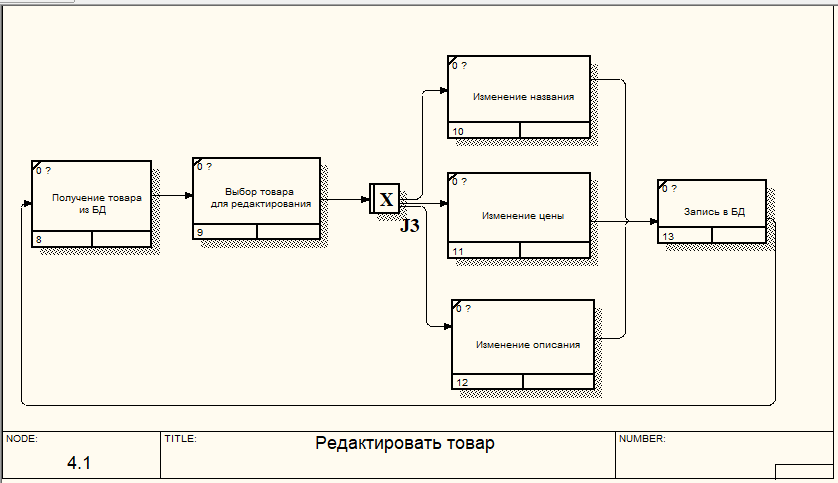


Рисунок 4 - Декомпозиция блока IDEF3

В ходе выполнения работы по моделированию предметной области были получены навыки создания функциональных моделей, контекстных диаграмм, диаграммы разложения. IDEF0 - методология представляет собой четко формализованный подход к созданию функциональных моделей - структурных схем изучаемой системы. Схемы строятся по иерархическому принципу с необходимой степенью подробности и помогают разобраться в том, что происходит в изучаемой системе, какие функции в ней выполняются и в какие отношения вступают между собой и с окружающей средой ее функциональные блоки. Совокупность схем (IDEF0 - диаграмм) образует модель системы. Эта модель носит качественный, описательный, декларативный характер.

Методология функционального моделирования IDEF0 является достаточно простым инструментом, который позволяет разработчикам корпоративных информационных систем изучить сферу деятельности заказчика и решать задачи по повышению эффективности этой деятельности.

# 2 Методология объектно-ориентированного моделирования

Ознакомление с основными элементами определения, представления, проектирования и моделирования программных систем с помощью языка UML.

UML - это унифицированный графический язык моделирования для описания, визуализации, проектирования и документирования ОО систем. UML призван поддерживать процесс моделирования ПС на основе ОО подхода, организовывать взаимосвязь концептуальных и программных понятий, отражать проблемы масштабирования сложных систем. Модели на UML используются на всех этапах жизненного цикла ПС, начиная с бизнес-анализа и заканчивая сопровождением системы. Разные организации могут применять UML по своему усмотрению в зависимости от своих проблемных областей и используемых технологий.

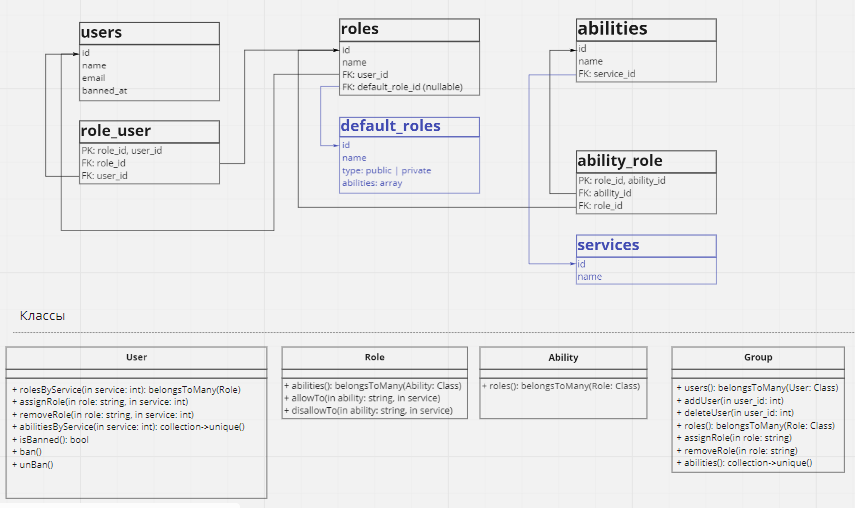


Рисунок 5 - Система ролей

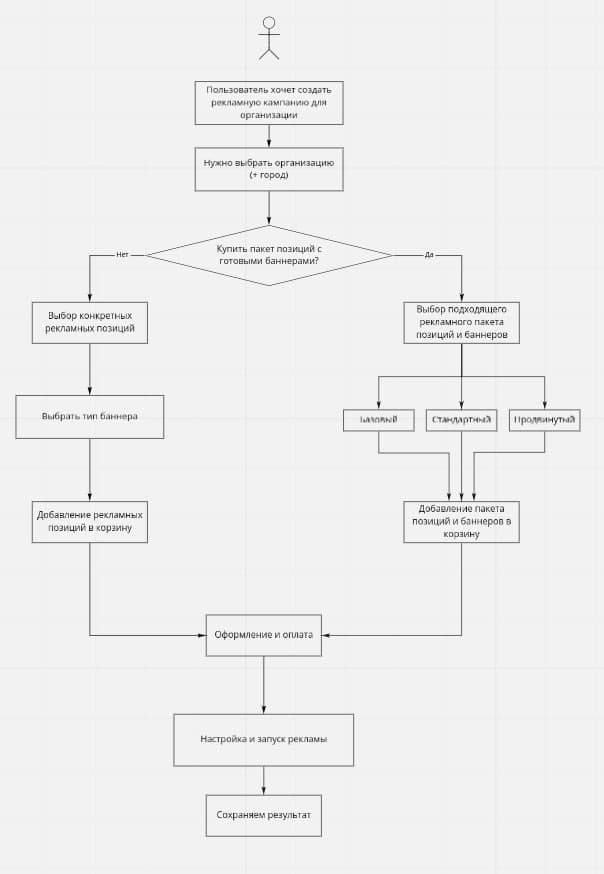


Рисунок 6 - Создание рекламной кампании

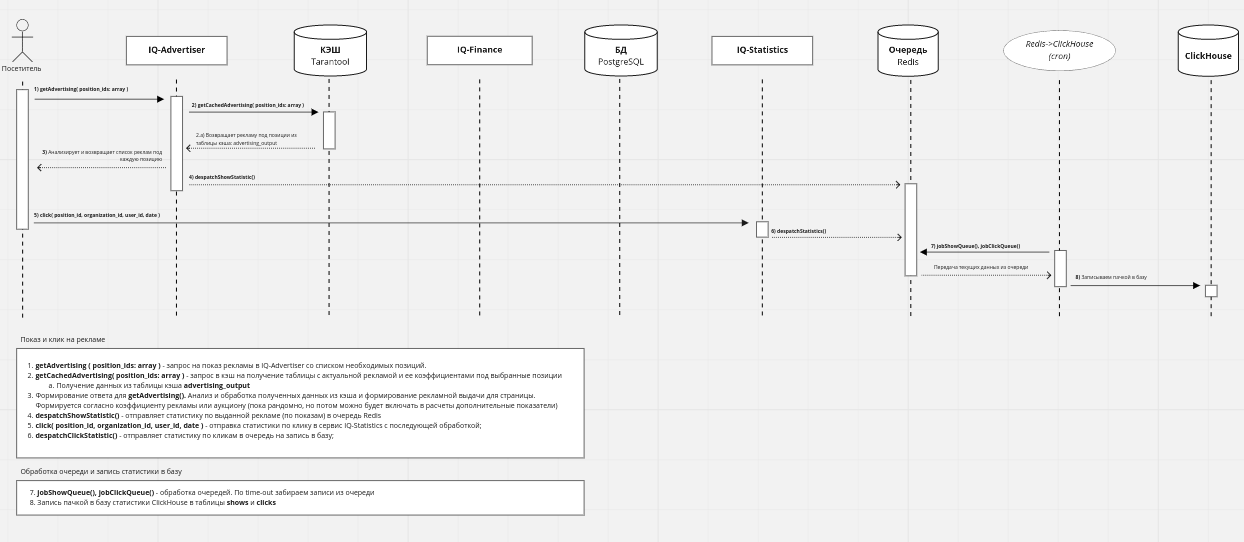


Рисунок 7 - Рекламная сеть

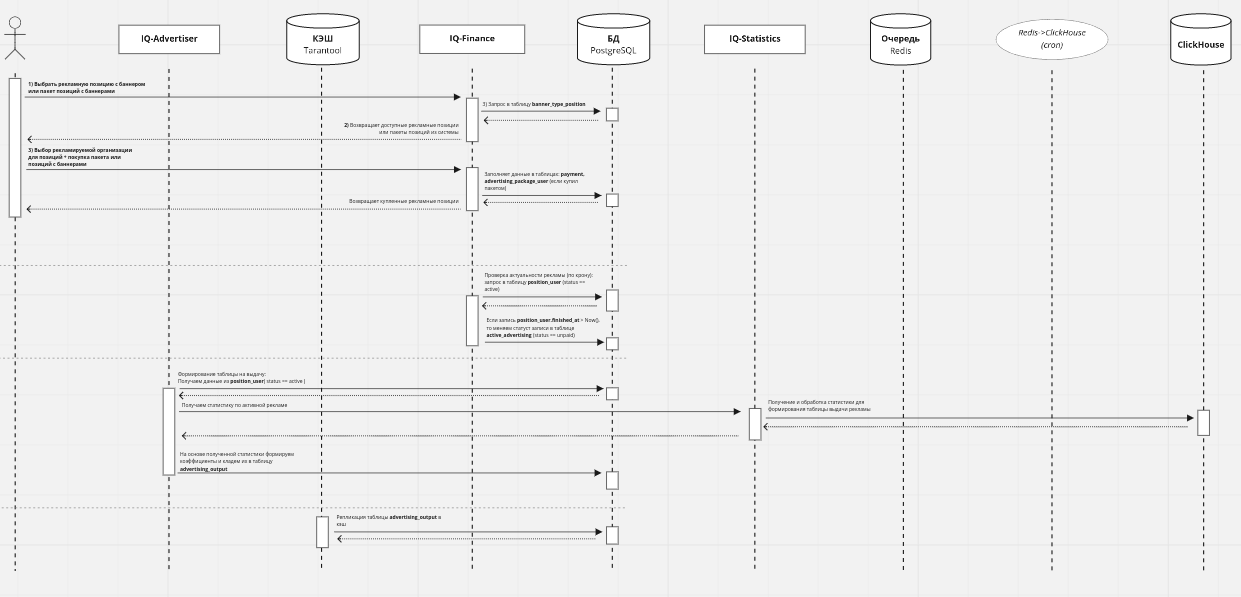


Рисунок 8 - Работа со статистикой

В ходе работы были получены навыки построения различных UML диаграмм, таких как диаграмма классов, диаграмма последовательностей, диаграмма прецедентов, диаграмма состояний. Диаграммы разных видов позволяют взглянуть на систему с разных точек зрения. Язык UML уже сейчас находит широкое применение в качестве неофициального стандарта в процессе разработки программных систем, связанных с такими областями, как моделирование бизнеса, управление требованиями, анализ и проектирование, программирование и тестирование. Применительно к этим процессам в языке UML унифицированы стандартные обозначения основных элементов соответствующих предметных областей.

# 3 Методология управления проектами

Данная работа направлена на ознакомление с основными понятиями методологии управления проектами, получение навыков по применению данных понятий при построении плана проекта, построения графика работ, управления рисками. Необходимо определить следующие этапы:

- определение всех этапов проекта, зависимых этапов, определение длительности этапов;

- построение на основе полученных данных сетевой и временной диаграмм;

- построение диаграммы распределения времени работы по этапам.

Таблица 1. Этапы выполнения работы

|  |  |  |
| --- | --- | --- |
| 1. | Бриф | 01.01 – 02.01 |
| 2. | Смета и договор | 02.01 – 05.01 |
| 3. | Проектирование | 05.01 – 15.01 |
| 4. | Дизайн | 10.01 – 15.01 |
| 5. | Верстка | 15.01 – 22.01 |
| 6. | Программирование | 22.01 – 29.01 |
| 7. | Тестирование | 29.01 – 03.02 |

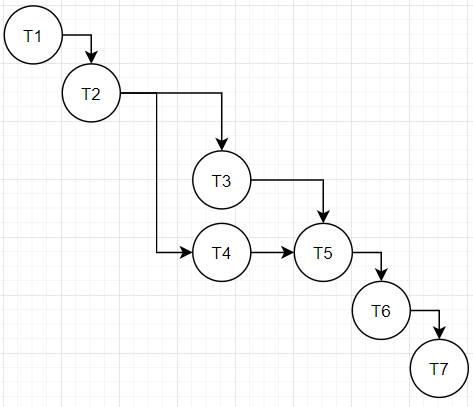


Рисунок 9 - Сетевая диаграмма этапов

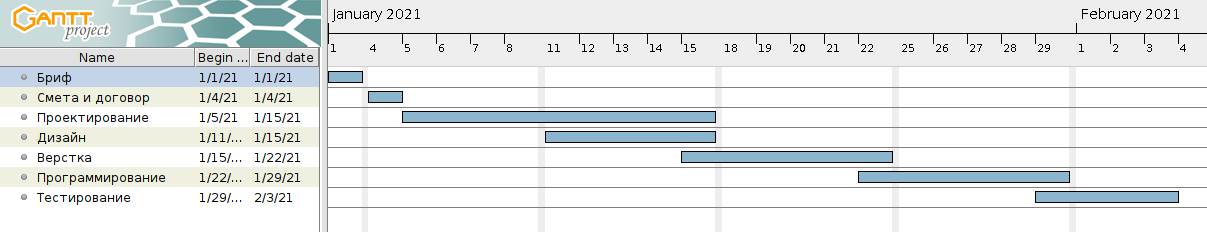


Рисунок 10 - Временная диаграмма длительности этапов

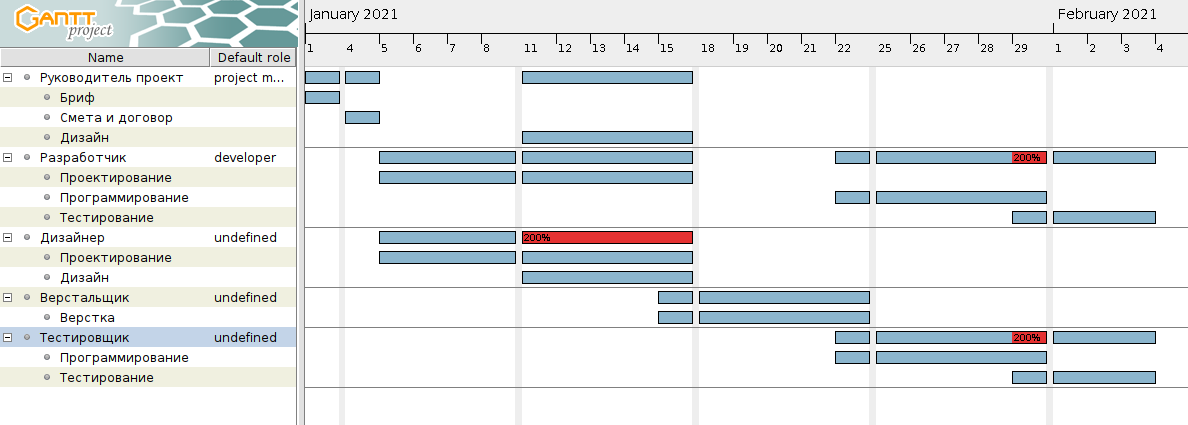


Рисунок 11 - Диаграмма загруженности персонала

Таблица 2. Риски

|  |  |  |
| --- | --- | --- |
| **Риск** | **Типы риска** | **Описание риска** |
| коммерческий риск | Риск для проекта | риск превышения выделенного на проект бюджета |
| отключение сервера | Риск для проекта | отсутствие источников бесперебойного питания и отсутствие системы резервного копирования, которая будет архивировать и автоматически копировать наработанные за день данные на независимых от работы сервера или дисководы массивов носителях |
| ошибки разработчиков | Риск для разрабатываемого продукта | критические ошибки разработчиков, приводящие к неработоспособности системы, связанные с недостаточным уровнем квалификации |
| взлом | Риск для разрабатываемого продукта | Получение доступа к системе злоумышленниками с целью перехвата клиентской базы |
| неопределенность требований заказчика | бизнес риск | некорректная формулировка требований заказчика, ведущая за собой не соответствие реализованной системы и желаемого |
| отклонение от сроков | бизнес риск | риск неправильного планирования, крайне сжатых сроков и неспособности разработчиков адаптироваться |

В качестве заключения нужно отметить необходимость осуществлять управление рисками на всех этапах работы. Мониторинг рисков также не обходим, как контроль топлива при поездке на автомобиле. Мониторинг рисков — это процесс отслеживания уже идентифицированных и поиск еще не выявленных рисков, а также оценки эффективности исполнения операций реагирования на риски. Мониторинг рисков включает в себя процедуры аудита и пересмотра рисков. Необходимо регулярно осуществлять пересмотр рисков. Совещания команд, осуществляющих разработку веб-проекта, должны включать в себя обсуждение процедур управления рисками. Аудит рисков обеспечивает оценку эффективности мероприятий по управлению на рисками.

# 4 Использование систем контроля версий исходного кода программ

В качестве системы контроля версий для реализуемого проекта была выбрана GIT. Система контроля версий — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии. Для контроля версий файлов в этой книге в качестве примера будет использоваться исходный код программного обеспечения, хотя на самом деле вы можете использовать контроль версий практически для любых типов файлов.

Система контроля версий даёт возможность возвращать отдельные файлы к прежнему виду, возвращать к прежнему состоянию весь проект, просматривать происходящие со временем изменения, определять, кто последним вносил изменения во внезапно переставший работать модуль, кто и когда внёс в код какую-то ошибку, и многое другое. Вообще, если, пользуясь системой контроля версий, вы всё испортите или потеряете файлы, всё можно будет легко восстановить.

Этапы работы с системой контроля версий:

1. Создания репозитория в Github.com
2. Инициализация системы Git в рабочей директории с проектом с помощью команды *git init*
3. Работа в проекте и фиксация изменений через систему коммитов, команда *git add* . – для добавления изменений и *команда git commit –m “comment*” – для коммита
4. Добавление репозитория через команду *git remote add*
5. Закачать коммит в репозитория через команду *git push*

На этом базовые команды для работы с системой версией Git закончены.

Для настройки веток можно использовать команду *git branch* c различными флагами.

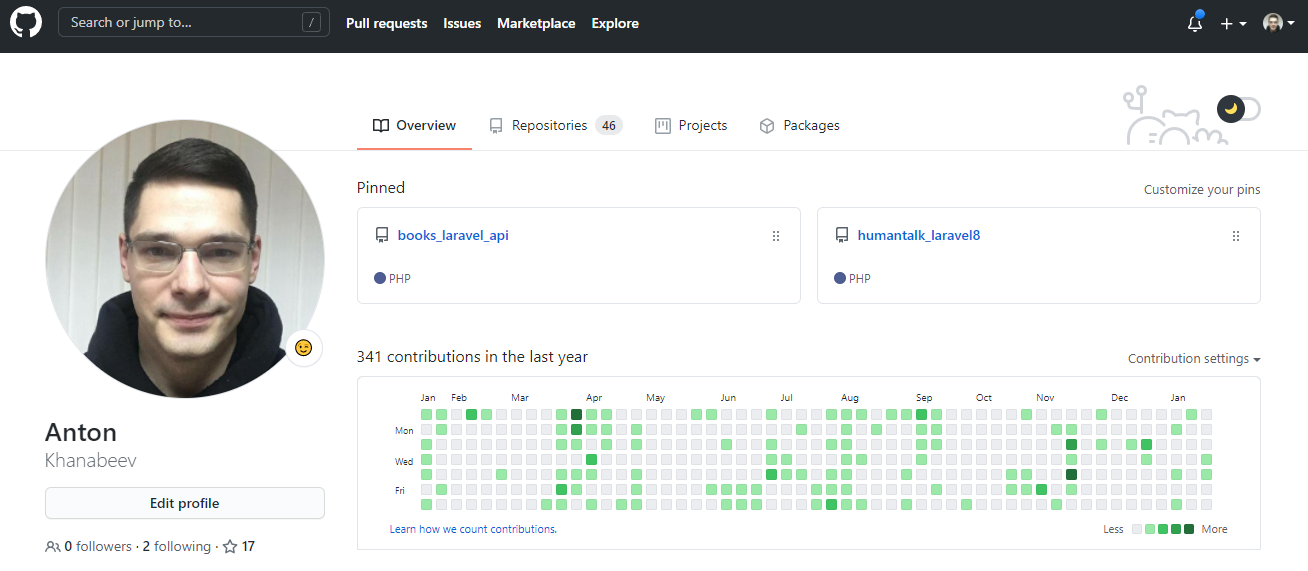


Рисунок 12 - Интерфейс сайта GitHub.com

**Список используемой литературы**

1. Мюллер, Роберт Дж. Проектирование баз данных и UML / Мюллер Роберт Дж.. - М.: ЛОРИ, 2013. - 422 c.
2. Роберт, А. Максимчук UML для простых смертных / Роберт А. Максимчук, Эрик Дж. Нейбург. - Москва: СИНТЕГ, 2014. - 272 c.
3. Рогозов Ю.И., Стукотий Л.Н., Свиридов А.С. «Моделирование систем» ТРТУ, 2004.
4. С.В. Маклаков «CASE-средства разработки информационных систем. BPwin и Erwin» –М.: ДиалогМифи, 2001.