

Главное управление образования Гродненского облисполкома
Учреждение образования
«Гродненский государственный политехнический колледж»

Специальность: 2 – 40 01 01 «Программное обеспечение информационных технологий»

Специализация: 2 – 40 01 01 33 «Компьютерная графика»

Дисциплина: «Технология разработки программного обеспечения»

Группа: ПЗТ – 35

Техно-рабочий проект

Тема: «Разработка сайта «Goying Beyond»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Разработал

Р.В.Хардигов

Руководитель проекта

О.В.Кизер

Содержание

Техническое задание.....	3
Требование к программе или программному изделию.....	3
Стадии и этапы разработки.....	5
Порядок контроля и приемки.....	7
Тестирование.....	7
Приложение А.....	18

					КП 2-40 01 01.33.35.14.22 ПЗ	Лист
						2
Изм.	Лист	№ докум.	Подпись	Дата		

Техническое задание

Наименование программы

Наименование программного продукта – сайт «goingbeyond.com»

Краткая характеристика области применения

Программа предназначена для публикации новостей, статей, и обсуждений игры Going Beyond.

Основания для разработки

Основанием для разработки является задание на практику на тему «сайт для Going Beyond». Тема согласована с преподавателем.

Назначение разработки

Основным назначением программы является изучение принципов создания программного обеспечения на языке программирования JavaScript, изучить как создается сайт, создание программной документации.

Требования к программе или программному изделию

Требования к применению

Устройство должно иметь постоянное подключение к Интернету.

Требования к производительности

- Для использования требуется версия ОС Android 6.0 и выше
- Windows 7, 10, 11

Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения следующих функций гостя:

- Просмотр новостей
- регистрация

Программа должна обеспечивать возможность выполнения следующих функций пользователя:

					КП 2-40 01 01.33.35.14.22 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

- авторизация
- комментирование
- предложение об добавлении/изменении информации
- участие в обсуждениях с другими пользователями

Программа должна обеспечивать возможность выполнения следующих функций администратора:

- авторизация
- просмотр предложений пользователей
- добавление/удаление информации на сайте
- модерация обсуждений пользователей

Требования к реализации

Для достижения максимальной производительности и оптимизации программа должна быть разработана на языке C# с использованием технологии WPF (Windows Presentation Foundation) в среде Visual Studio 2019.

Требование к надежности

- Сайт должен иметь защиту данных пользователей.
- Администратором не может стать любой человек.
- Сайт не может быть изменён никем, кроме разработчика и администратора

Требования к интерфейсу

Интерфейс должен быть интуитивно понятным, должен быть с приятной цветовой гаммой.

Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователя было просто и понятно принимать информацию.

Пользователь не должен испытывать какого-либо дискомфорта в плане восприятия информация, отображённой на экране. Объекты (рисунки и символы) не должны быть слишком мелкие.

Слева будет значок для открытия выдвижного меню. В меню будут

					КП 2-40 01 01.33.35.14.22 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

присутствовать основные вкладки, поиск, переход на страницу пользователя.

Требования к программной документации

Для удобства использования программного продукта пользователем, необходимо добавить файл справки, в котором будет находиться вся необходимая информация о работе с сайтом.

Технико-экономические показатели

Ориентировочная экономическая эффективность не рассчитываются. Предполагаемое число использования программой в год – 365 сеансов на одном рабочем месте.

Стадии и этапы разработки

Стадии разработки

Разработка должна быть проведена в три стадии: разработка технического задания, рабочее проектирование, внедрение.

Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

разработка программы;

разработка программной документации;

испытания программы.

На стадии внедрения должен быть выполнен этап разработки – подготовка и передача программы.

					КП 2-40 01 01.33.35.14.22 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. Постановка задачи;
2. Определение и уточнение требований к техническим средствам;
3. Определение требований к программе;
4. Определение стадий, этапов и сроков разработки программы и документации на неё;
5. Выбор языков программирования;
6. Согласование и утверждение технического задания.

На этапе разработки программной документации должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 с требованием п. Предварительный состав программной документации настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

Разработка, согласование и утверждение программы и методики испытаний;

Проведение приемо-сдаточных испытаний;

Корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика. Порядок контроля и приемки.

					КП 2-40 01 01.33.35.14.22 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Виды испытаний

Приемо-сдаточные испытания программы должны проводиться согласно разработанной Исполнителем и согласованной Заказчиком программы и методик испытаний.

Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний.

Общие требования к приемке работы

На основе Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывают Акт приемки-сдачи программы в эксплуатацию.

Тестирование

Таблица 1-Стратегии тестирования

№	Название стратегии	Сущность	Преимущества	Недостатки	Кто осуществляет
1	Метод белого ящика	Метод тестирования ПО, который предполагает, что внутренняя структура/устройство/реализация системы известны тому, кто её тестирует.	<ul style="list-style-type: none">- Оптимизация кода путем нахождения скрытых ошибок- Доступность структуры кода позволяет выбрать тип входных данных, необходимых для эффективного тестирования- Возможность автоматизирования тест-кейсов	<ul style="list-style-type: none">- Поскольку знание кода и внутренней структуры является необходимым условием, для проведения такого тестирования требуется квалифицированный тестировщик, что увеличивает стоимость- И почти невозможно изучить каждый кусок кода, чтобы обнаружить скрытые ошибки, что может создать проблемы, приводящие к сбою приложения	Метод белого ящика часто используется на стадии, когда приложение еще не собрано воедино, но необходимо проверить каждый из его компонентов, модули процедур и подпрограмм. Компонентным тестирование чаще всего занимается программист, хорошо понимающий код или тестировщик, имеющий прекрасные знания в области программирования.

Таблица 1 - продолжение

2	Метод чёрного ящика	Это стратегия, в которой тестирование основано исключительно на требованиях и спецификациях, при этом мы не знаем, как устроена внутри тестируемая система и работаем исключительно с внешними интерфейсами тестируемой системы или компонента.	<ul style="list-style-type: none"> - Простота: облегчает тестирование проектов высокого уровня и сложных приложений - Экономия ресурсов: тестеры сосредоточены на функциональности программного обеспечения - Тестовые случаи: Сосредоточение внимания на функциональности программного обеспечения для облегчения быстрой разработки тестовых случаев - Обеспечивает гибкость: специальные знания программирования не требуются 	<ul style="list-style-type: none"> - В действительности выполняется избранное число тестовых сценариев, результатом чего является ограниченный охват - Отсутствие четкой спецификации затрудняет разработку тестовых сценариев - Низкая эффективность 	При тестировании по стратегии чёрного ящика исследуются все отдельные компоненты, например, интерфейс пользователя и UX, веб-сервер или сервер приложения, база данных, зависимости и интегрированные системы.
3	Метод серого ящика	Это метод тестирования программного обеспечения для тестирования программного продукта или приложения с частичным знанием внутренней структуры приложения. Целью тестирования серого ящика является поиск и выявление дефектов, вызванных неправильной структурой кода или неправильным использованием приложений.	<ul style="list-style-type: none"> - Позволяет быстро выявить ошибки в функциональных спецификациях - Тестировщику не нужна дополнительная квалификация - Тестирование проходит «с позиции» пользователя - Составлять тест-кейсы можно сразу после подготовки спецификации 	<ul style="list-style-type: none"> - Связывание дефектов затруднено, когда тестирование серого выполняется для распределенных систем - Ограниченный доступ к внутренней структуре приводит к ограниченному доступу для обхода пути кода - Поскольку доступ к исходному коду невозможен, полное тестирование белого ящика невозможно 	Тестирование серого ящика - это продукт тестирования черного ящика и тестирования белого ящика. Black Box Testing означает, что тестер не знает, как работает программное обеспечение внутри. Этот тип тестирования выполняется на уровне пользователя. Таким образом, тестер проверяет, получен ли конечный результат, и не знает, правильно ли работает код в циклах и разрывах внутри. Таким образом, именно тестировщики программного обеспечения отвечают за тестирование черного ящика.

Таблица 2-Уровни тестирования

№	Название уровня	Сущность	Когда осуществляется	Кто осуществляет
1	Модульное	Уровень тестирования, на котором тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция.	Чтобы проверить, правильно ли написан модуль, проводят юнит-тесты, или модульное тестирование, — проверку не всего приложения, а одного модуля. Главная причина написания юнит-тестов — тестирование отдельных модулей.	Модульные тесты — это, как правило, автоматические тесты, написанные и выполняемые разработчиками программного обеспечения для обеспечения того, чтобы раздел приложения соответствовал его дизайну и вел себя должным образом.
2	Интеграционное	Это этап тестирования программного обеспечения, на котором отдельные программные модули объединяются и тестируются как группа.	Целью нашего тестирования является выявление багов при взаимодействии между этими программными модулями и в первую очередь направлен на проверку обмена данными между этими самими модулями.	Как правило, программный продукт состоит из нескольких программных модулей, написанных разными программистами
3	Системное	Системное тестирование означает тестирование всей системы в целом, оно выполняется после интеграционного тестирования, чтобы проверить, работает ли вся система целиком должным образом.	Системное тестирование выполняется для всей системы в контексте либо спецификаций функциональных требований (FRS), либо спецификаций системных требований (SRS), либо и того, и другого.	Поскольку системное тестирование - процесс, требующих значительных ресурсов, для его проведения часто выделяют отдельный коллектив тестировщиков, а зачастую системное тестирование выполняется организацией, которая не связана с коллективом разработчиков и тестировщиков, выполнявших работы на предыдущих этапах тестирования
4	Приемочное	Это комплексное тестирование, необходимое для определения уровня готовности системы к последующей эксплуатации.	Оно проводится после тестирования системы и является последним этапом процесса тестирования программного обеспечения.	Иногда приемочное тестирование выполняет специальная группа тестирования, включающая представителей конечных пользователей. В других случаях приемочное тестирование выполняется группой, состоящей только из представителей заказчика или уполномоченных им.
5	Выходное	Осуществляется с целью проверки готовности программного обеспечения для поставки заказчику/пользователям.	Выходное тестирование осуществляется с целью проверки готовности программного обеспечения для поставки заказчику/пользователям.	Это завершающий этап тестирования, проводимый независимым тестировщиком, включающий в себя проверку на корректность инструкций по установке, а также проверку комплектности документации.

Таблица 3-Виды тестирования

№	Название вида	Сущность
1	По запуску кода на исполнение: 1.1 Статическое 1.2 Динамическое	<p>1.1 Оно представляет собой процесс или технику, которые выполняются для поиска потенциальных дефектов в программном обеспечении. Это также процесс обнаружения и устранения ошибок и дефектов в различных сопроводительных документах</p> <p>1.2 Основная идея этого вида тестирования состоит в том, что проверяется реальное поведение (части) приложения. Проще говоря, динамическое тестирование выполняется путем фактического использования приложения и определения того, работает ли функциональность так, как ожидается. Динамическое тестирование включает в себя тестирование ПО в режиме реального времени путем предоставления входных данных и изучения результата поведения программы. Проверка осуществляется с помощью ручного или автоматического выполнения заранее подготовленного набора тестов. Оно является частью процесса валидации программного обеспечения.</p>
2	Функциональные виды 2.1 Функциональное 2.2 Тестирование безопасности 2.3 Тестирование взаимодействия	<p>2.1 Это вид тестирования, при котором выявляется некорректная /неправильная работа функционала программы. Проверка функций и характеристик разрабатываемого ПО. Этот вид тестирования занимает 90% времени, отведённого на тестирование. Функциональное тестирование предполагает проверку функциональных требований: логики и бизнес-правил приложения или системы.</p> <p>2.2 Тестирование безопасности — это процесс, направленный на выявление недостатков в механизмах безопасности информационной системы, которые защищают данные и поддерживают функциональность по назначению. Из-за логических ограничений тестирования безопасности прохождение процесса тестирования безопасности не является признаком того, что никаких недостатков не существует или что система адекватно удовлетворяет требованиям безопасности.</p>

Талица 3 – продолжение

					<p>2.3 Тестирование взаимодействия – это функциональное тестирование, проверяющее способность приложения взаимодействовать с одним и более компонентами или системами и включающее в себя тестирование совместимости, и интеграционное тестирование.</p> <p>Программное обеспечение с хорошими характеристиками взаимодействия может быть легко интегрировано с другими системами, не требуя каких-либо серьезных модификаций.</p>
3	<p>Нефункциональные виды:</p> <p>3.1 Тестирование производительности (нагрузочное тестирование, стрессовое тестирование, тестирование стабильности или надежности, объемное тестирование)</p> <p>3.2. Тестирование установки</p> <p>3.3. Тестирование удобства пользования</p> <p>3.4. Тестирование на отказ и восстановление</p> <p>3.5. Конфигурационное тестирование</p>				<p>3.1 Тестирование, которое проводится с целью определения, как быстро работает вычислительная система или её часть под определённой нагрузкой. Также может служить для проверки и подтверждения других атрибутов качества системы, таких как масштабируемость, надёжность и потребление ресурсов.</p> <p>3.2 Тестирование установки может быть направлено на поиск ошибок, возникающих в процессе установки, которые влияют на восприятие пользователем и способность использовать установленное программное обеспечение. Существует множество событий, которые могут повлиять на установку программного обеспечения, и тестирование установки может проверять правильность установки, одновременно проверяя ряд связанных действий и событий.</p> <p>3.3 Это метод оценки интерфейса со стороны удобства и эффективности его использования. Чаще всего оно проводится, когда у страницы, сайта, приложения низкая посещаемость или есть жалобы на проблемы в работе интернет-ресурса. Например, пользователи пишут в поддержку: «Я не могу оформить заказ, товар не отображается в корзине», и мы проводим тестирование, чтобы разобраться с этой жалобой.</p> <p>3.4 Подвид тестирования производительности, проверяет тестируемый продукт с точки зрения способности противостоять и успешно восстанавливаться после возможных сбоев, возникших в связи с ошибками ПО, отказами оборудования или проблемами связи/сети.</p> <p>3.5 специальный вид тестирования, направленный на проверку работы ПО при различных аппаратных и программных конфигурациях системы</p>
Изм.	Лист	№ докум.	Подпись	Дата	<p>КП 2-40 01 01.33.35.14.22 ПЗ</p> <p>Лист</p> <p>11</p>

Таблица 3 – продолжение

4	<p>Связанные с изменениями</p> <p>4.1. Дымовое тестирование</p> <p>4.2. Регрессионное тестирование</p> <p>4.3. Тестирование сборки</p> <p>4.4. Санитарное тестирование или проверка согласованности/исправности</p>	<p>4.1 В тестировании программного обеспечения означает минимальный набор тестов на явные ошибки. Дымовой тест обычно выполняется программистом; не прошедшую этот тест программу не имеет смысла отдавать на более глубокое тестирование.</p> <p>4.2 Это повторный запуск функциональных и нефункциональных тестов для обеспечения того, чтобы ранее разработанное и протестированное программное обеспечение по-прежнему работало после изменения. Если нет, то это называется регрессией.</p> <p>4.3 Это тестирование, направленное на определение соответствия, выпущенной версии, критериям качества для начала тестирования. По своим целям является аналогом Дымового Тестирования, направленного на приемку новой версии в дальнейшее тестирование или эксплуатацию. Вглубь оно может проникать дальше, в зависимости от требований к качеству выпущенной версии.</p> <p>4.4 Это узконаправленное тестирование достаточное для доказательства того, что конкретная функция работает согласно заявленным в спецификации требованиям. Является подмножеством регрессионного тестирования. Используется для определения работоспособности определенной части приложения после изменений, произведенных в ней или окружающей среде. Обычно выполняется вручную.</p>
5	<p>По степени автоматизации:</p> <p>5.1. Ручное тестирование</p> <p>5.2. Автоматизированное тестирование</p>	<p>5.1 часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения. Оно производится тестировщиком без использования программных средств, для проверки программы или сайта путём моделирования действий пользователя. В роли тестировщиков могут выступать и обычные пользователи, сообщая разработчикам о найденных ошибках.</p> <p>5.2 Процесс тестирования программного обеспечения, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, производятся автоматически с помощью инструментов для автоматизированного тестирования.</p>

Талица 3 – продолжение

6	По времени проведения: 6.1. Альфа-тестирование 6.2. Бета-тестирование	<p>6.1. Это вид приемочного тестирования, которое обычно проводится на поздней стадии разработки продукта и включает имитацию реального использования продукта штатными разработчиками либо командой тестировщиков. Обычно альфа тестирование заключается в систематической проверке всех функций программы с использованием техник тестирования «белого ящика» и «черного ящика»</p> <p>Альфа-тестирование является методологией оценки качества и стабильности тестируемого продукта в тестовой среде.</p> <p>6.2. Эксплуатационное тестирование потенциальными и/или существующими клиентами/заказчиками на внешней стороне, никак не связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы.</p>
---	---	---

Таблица 1-Тесты на использование

№	Наименование функции	Описание тестового сценария с исходными данными	Ожидаемый результат
1	Переход в соцсеть разработчика	1.В подвале главной страницы расположена кнопка социальной сети ведущей на страницу разработчика.	Переход в социальную сеть
2	Быстрое перемещение по сайту	1.На шапке сайта находятся кнопка «Зарегистрироваться» Нажимаем на неё, куда хотим перейти	Быстрый переход к выбранному разделу
3	Поделиться в соцсети	1.Нажать на иконку соцсети и поделиться	Ссылка на новость отправлена выбранному получателю
4	Подняться вверх страницы по кнопке	1.При прокрутке сайта справа появляется кнопка, позволяющая быстро вернуться в начало страницы	Возврат в начало страницы
5	Отправка комментариев	1. Заполняем обязательные поля "Email", "Ваш комментарий". 2.Нажатие на кнопку «Отправить»	Комментарий отправлен
6	Возврат на главную страницу с страницы новости	1. В шапке сайта нажать кнопку «Вернуться на главную»	Возврат к главной странице
7	Отправить данные для регистрации	1. Заполняем обязательные поля "Email", "Имя пользователя" "Пароль". 2. Нажимаем на кнопку зарегистрироваться	Данные отправляются для регистрации

Приложение А

					КП 2-40 01 01.33.35.14.22 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

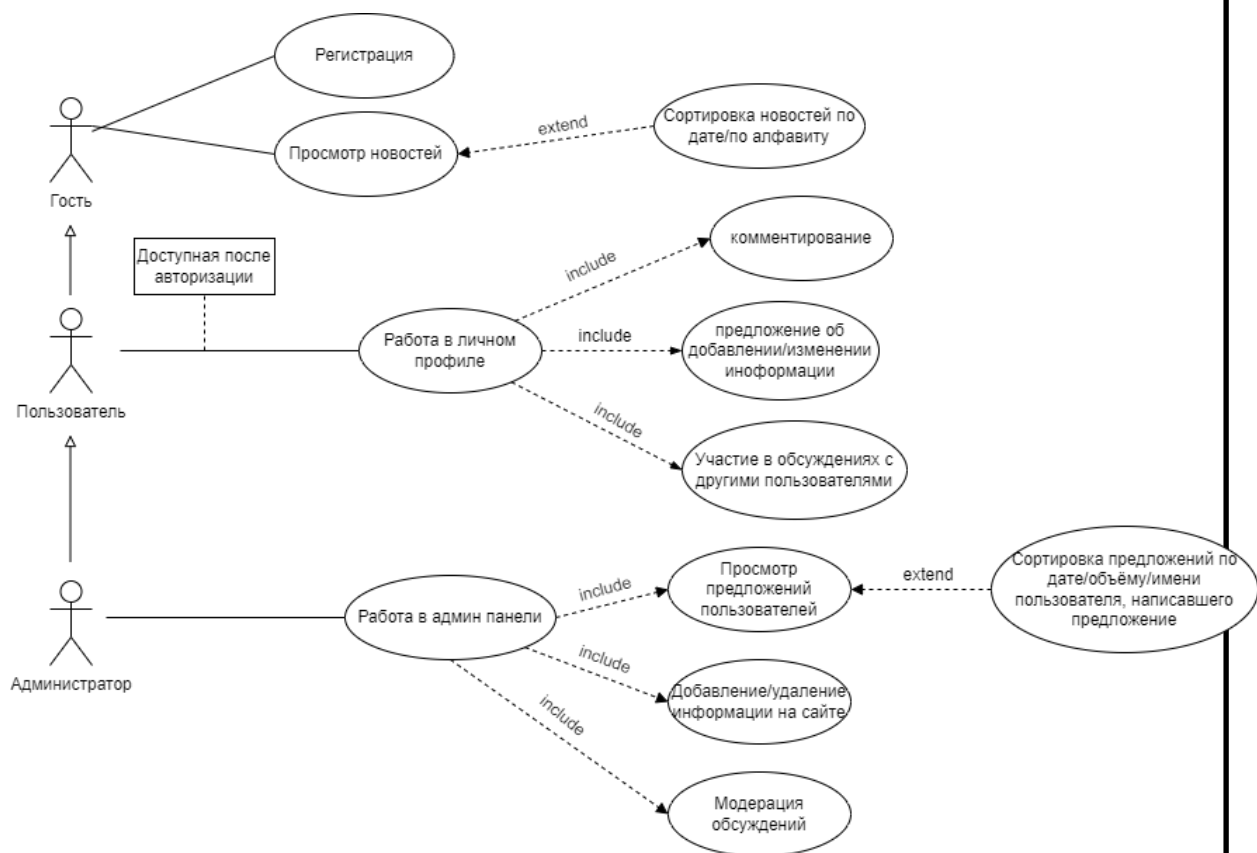


Рисунок 1 — Диаграмма вариантов использования

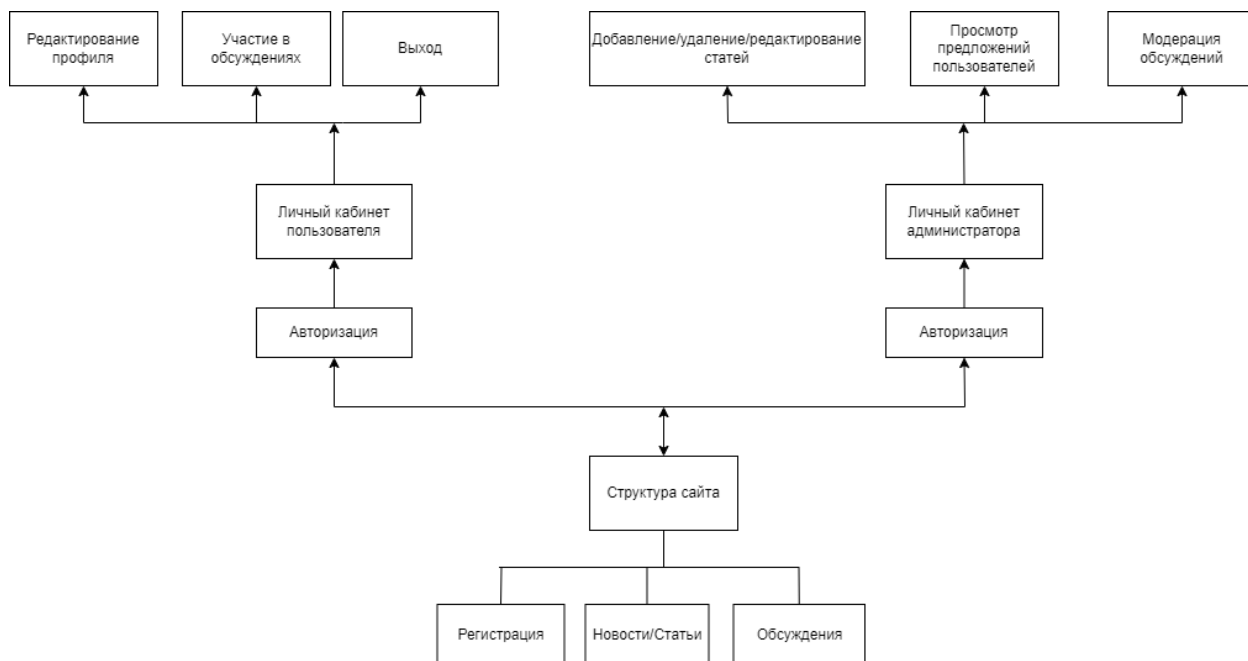


Рисунок 2 — Структура приложения



Рисунок 3 — Модель данных

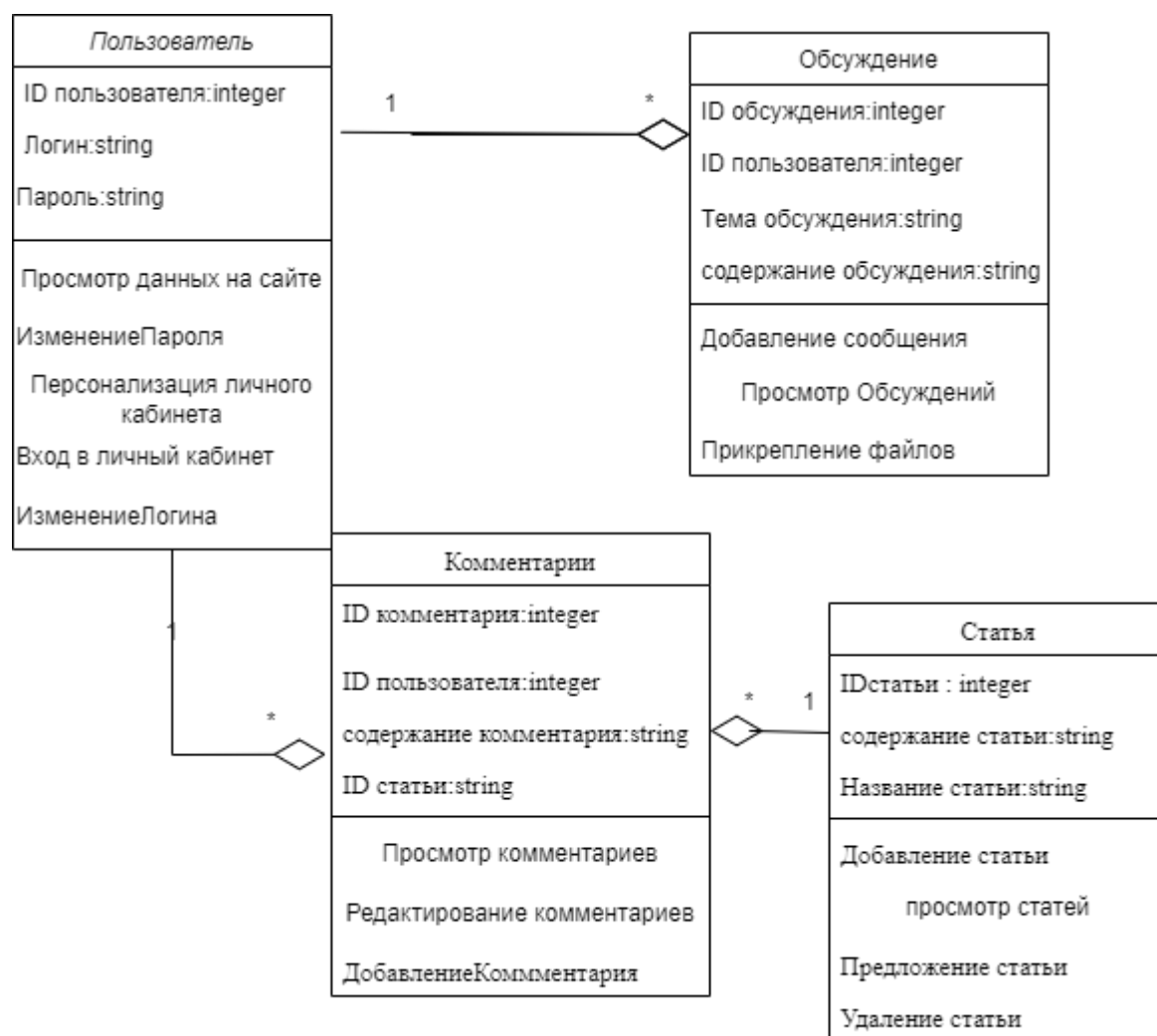


Рисунок 4 — Диаграмма классов



Рисунок 5 — Функциональная модель



Рисунок 6 — Функциональная модель (детализация)



Рисунок 7 — Диаграмма последовательности

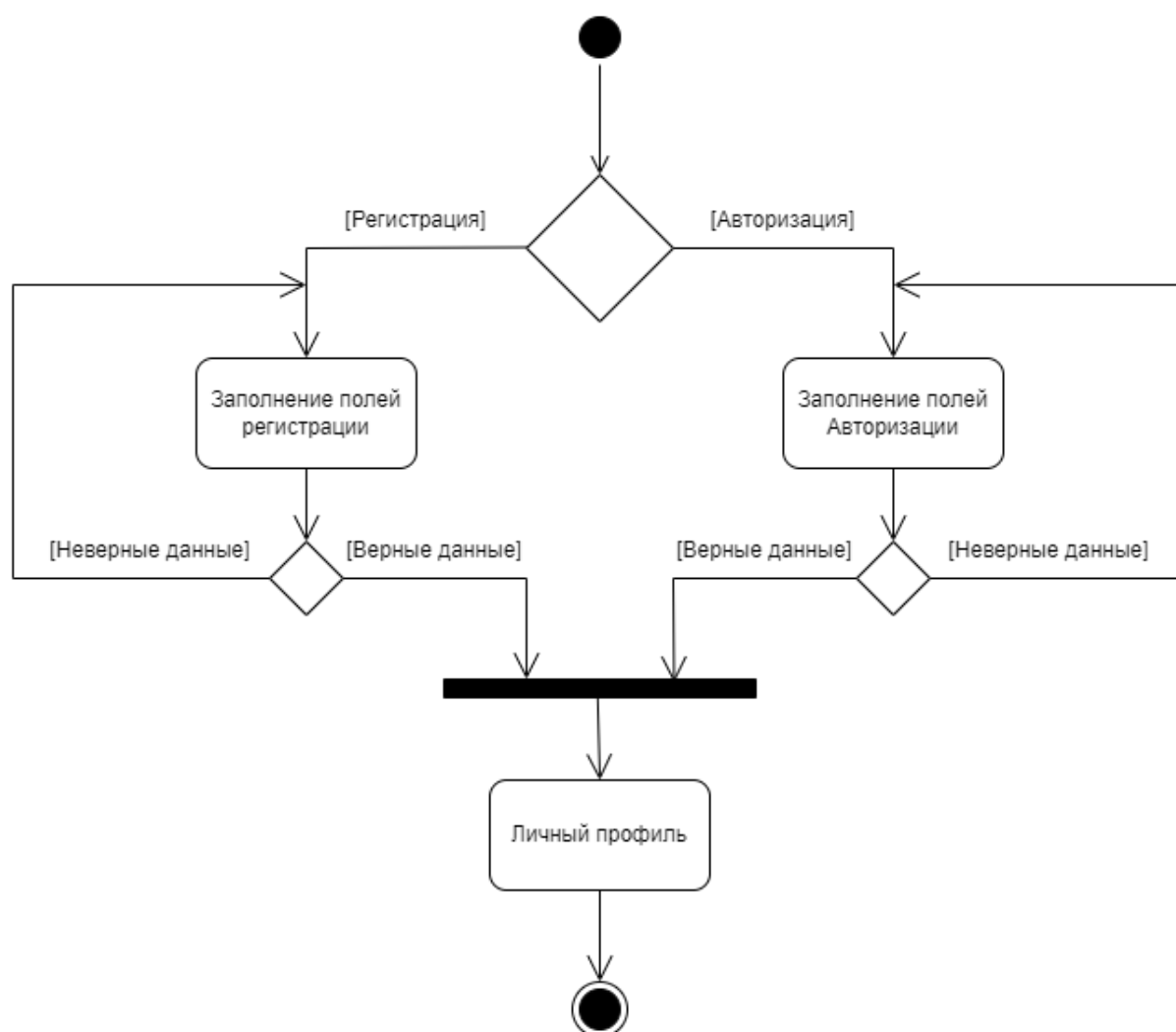


Рисунок 8 — Диаграмма деятельности

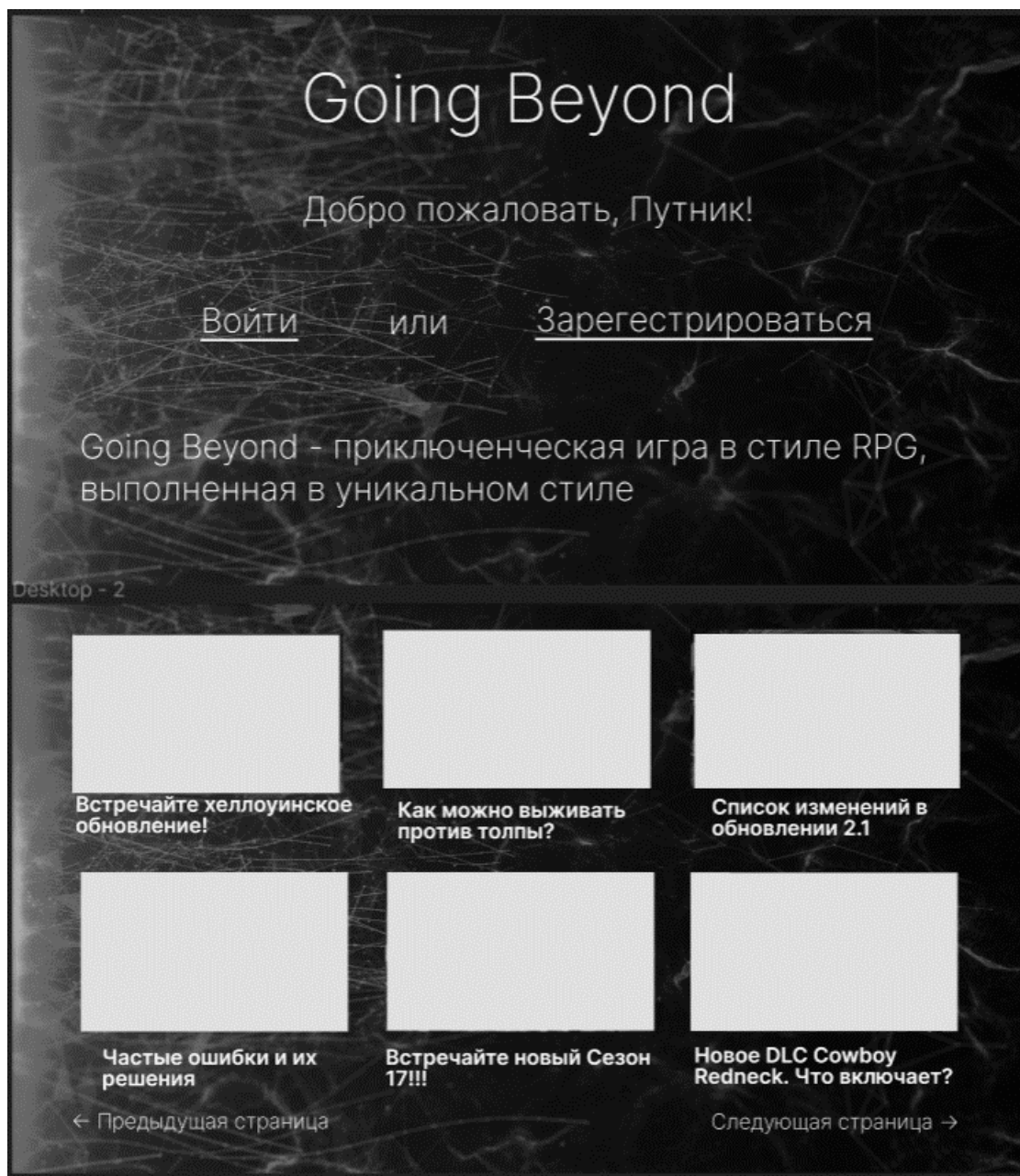


Рисунок 9 – Главная страница

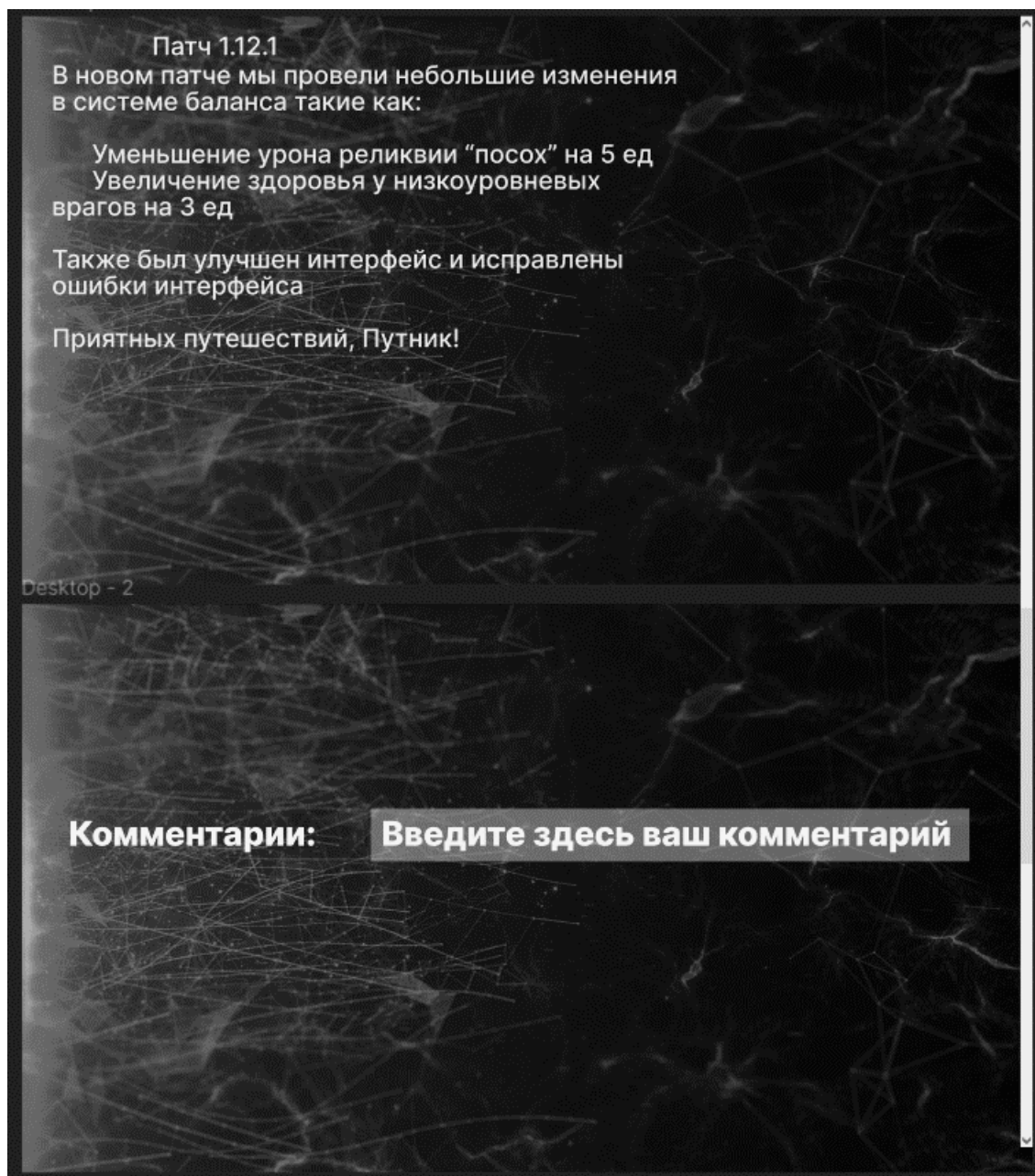


Рисунок 10 – страница новости



Рисунок 11 – Личный кабинет пользователя