Федеральное агентство по образованию РФ

ГОУ ВПО Нижегородский государственный университет им. Н.И. Лобачевского

Факультет Вычислительной математики и кибернетики

Кафедра Математического обеспечения ЭВМ

УЧЕБНЫЙ КУРС

**«Проектирование и архитектура программных систем»**

для подготовки по направлению «Программная инженерия»

**КОНЦЕПЦИЯ ПРОЕКТА**

**CreateHub**

Нижний Новгород  
2025

**Содержание[[1]](#footnote-0)**

[1.](#_heading=h.t215dda6x97k) Необходимость проекта 3

[1.1.](#_heading=h.neiaqrrk1kdk) Обоснование необходимости 3

[1.2.](#_heading=h.upv5y2ci3nau) Видение проекта 3

[1.3.](#_heading=h.emcdksfsgnub) Анализ выгод 4

[2.](#_heading=h.a2zl7pwj4xk2) Концепция решения 4

[2.1.](#_heading=h.j1sgwp7d3fhw) Цели и Задачи 4

[2.2.](#_heading=h.i8hwik4pzegd) Предположения и Ограничения 4

[2.3.](#_heading=h.j2xfp26b8w3f) Анализ использования 4

[2.3.1.](#_heading=h.8dbeu9zfhjyn) Пользователи 4

[2.3.2.](#_heading=h.63vkrjpmzvl2) Сценарии использования 5

[2.4.](#_heading=h.9xi8uy9tivqf) Требования 5

[2.4.1.](#_heading=h.u85b3wze728z) Требования пользователей 5

[2.4.2.](#_heading=h.3vh339v281oa) Системные требования 5

[3.](#_heading=h.5na7sryb6ajk) Рамки 5

[3.1.](#_heading=h.wn62bvd14s63) Функциональность решения 6

[3.2.](#_heading=h.b8m6iowip24t) За рамками решения 6

[3.3.](#_heading=h.23bdmoi7y44h) Критерии одобрения решения 6

[4.](#_heading=h.wv2u7hh1482q) Стратегии дизайна решения 6

[4.1.](#_heading=h.3fb4h0yd61s0) Стратегия архитектурного дизайна 6

[4.2.](#_heading=h.9cfw8nnok07w) Стратегия технологического дизайна 6

**Необходимость проекта**

**Обоснование необходимости**

Пользователи сталкиваются с проблемой сложности иллюстрации своего проекта ( комикса, манги, визуальной новеллы), тк не умеют рисовать или же не могут позволить нанять себе художника, который проиллюстрировал его работу.

**Видение проекта**

*Видение* (*vision*) – это ничем не ограничиваемое представление о том, каким должно быть *решение* (*solution*). Видение проекта направлено на формирование у всех вовлеченных в проект сторон единого понимания его концепции. *Формулировка видения* (*vision statement*) должна быть достаточно краткой для запоминания, достаточно ясной для понимания и достаточно сильной для мотивирования. Хорошая формулировка видения ориентируется на пять SMART характеристик:

*Specific* (определенность/конкретность) – видение четко указывает на то (идеальное) состояние, достижение которого является целью проекта.

*Measurable* (измеримость) – дает проектной группе четкий критерий успешности проекта и достижения поставленных целей.

*Achievable* (достижимость) – цели, сформулированные в видении, должны быть достижимы в рамках имеющихся ресурсов, времени и возможностей команды. Достижимость мотивирует команду на выполнение проекта.

*Relevant* (обоснованность) – цели, сформулированные в видении, должны иметь существенное значение для заинтересованных сторон и напрямую быть связанными с их проблемами и/или потребностями.

*Time-based* (ограниченность во времени) – видение должно четко указывать на ожидаемые временные рамки, в которые решение будет достигнуто.

Создан сайт, в котором пользователь может загружать в ИИ изображение своего персонажа. ИИ в свою очередь генерирует этого персонажа в разных позах и с разными эмоциями.

Пользователь может выбрать шаблон и изменять его в редакторе, а именно

* Выбирать задний фон
* Вставлять текст в специальных полях
* Вставлять своего персонажа

Все фоны и шаблоны пользователь выбирает из предложенного нами списка.

Так же будет разработан телеграм бот, который позволяет просматривать работы и создавать новых персонажей с помощью ИИ

Критерии оценивания:

* ИИ должна генерировать более 50 вариаций персонажа в разных позах и настроениях
* Работоспособность ключевых функций: загрузка изображений, генерация изображений ИИ, создание иллюстраций с помощью конструктора.
* Количество шаблонов и фонов не менее 10
* Пользователь так же может скачать свою работу в формате pdf.

MVP проекта должен быть сделан в течении этого семестра.

**Анализ выгод**

* Возможность визуализации проекта без навыков рисования
* ИИ позволяет создавать своего персонажа что увеличивает свободу творчества
* Не нужно продумывать задние фоны и шаблоны, они уже готовы

**Концепция решения**

Концепция решения (*solution concept*) предоставляет общее описание подходов, которые проектная группа предполагает использовать для разрешения проблем и/или удовлетворения потребностей заинтересованных сторон.

**Цели и Задачи**

Формирование концепции решения начинается с выяснения у заинтересованных сторон, описания и фиксации проектной группой целей проекта. Далее каждая цель разбивается на измеримые компоненты – задачи.

**Цель 1:** Регистрация пользователя и хранение его проектов

* Регистрация
* Авторизация
* Реализация личного кабинета, в котором пользователь может просматривать свои работы и своих персонажей

**Цель 2:** Генерация персонажей с помощью ИИ

* Пользователь может загрузить исходное изображение и/или ввести его описание
* Подключение готовой модели ИИ

**Цель 3:** Редактор

* Хранение фонов и шаблонов в БД
* Возможность пользователю выбирать шаблон
* Возможность выбирать фон
* Возможность вставки текста в специальные места в шаблоне
* Возможность вставки персонажа
* Возможность сохранения иллюстрации

**Цель 4:** Телеграмм бот

* Возможность создания персонажа с помощью ИИ
* Возможность просмотр своих работ

**Предположения и Ограничения**

В процессе формирования концепции решения проектная группа постоянно взаимодействует с заинтересованными сторонами, собирая необходимую информацию о требованиях к функциональности будущего решения. Тем не менее, неизбежная неполнота информации приводит к тому, что относительно некоторых функциональных возможностей решения могут потребоваться *предположения* (*assumptions*). Помимо функциональных требований заинтересованные стороны могут выдвигать качественные требования, задающие *ограничения* на создаваемое решение. Также ограничения могут порождаться средой, в которой должно будет функционировать решение после внедрения.

**Предположения:** Пользователи имеют стабильный интернет и браузер поддерживающий работу с графикой

**Ограничения:** Количество шаблонов и фонов ограничено

**Анализ использования**

Основой формулировки требований является анализ использования, включающий определение *пользователей* (*users*) и описание того, как пользователи будут взаимодействовать с решением.

**Пользователи**

В разработке решения заинтересованы множество сторон, однако непосредственная работа с ним будет выполняться пользователями, поэтому прежде чем приступать к дизайну решения, необходимо определить, кто будет с ним взаимодействовать. В процессе анализа должны быть выделены группы пользователей (например, на основе областей их деятельности, в которых будет использоваться разрабатываемое решение).

Авторы визуальных новелл, манги, комиксов, пользователи не умеющие рисовать.

**Сценарии использования**

*Сценарии использования* (*usage scenarios*) определяют последовательности действий, которые пользователи выполняют при взаимодействии с решением. Один из возможных (и достаточно распространенных) вариантов – язык UML.

**Сценарий:** "Регистрация".

Добавить uml диаграмму

**Сценарий:** "Создание персонажей".

Добавить uml диаграмму

**Сценарий:** "Работа с редактором".

Добавить uml диаграмму

**Сценарий:** "Работа с телеграмм ботом".

Добавить uml диаграмму

**Требования**

*Требования* (*requirements*) определяют, что должно делать разрабатываемое решение. Требования могут выражаться в терминах функциональности или в виде правил и параметров, определяющих функциональность.

**Требования пользователей**

Функциональные:

* Возможность создавать своего персонажа
* Возможность создавать/редактировать/удалять свой проект
* Возможность выбирать шаблоны и фоны и положение и настроение персонажа
* Возможность скачать свой проект в формате pdf

Нефункциональные:

* Ненагруженный интерфейс понятный человеку без опыта рисования
* Данные пользователя должны безопасно хранится и передаваться

**Системные требования**

Веб-браузер с поддержкой WebSocket, сервер на Django, СУБД PostgreSQL.

**Рамки**

*Рамки* (*scope*) определяют пространство параметров, в котором будет создаваться решение, детализируя функциональность, определяя, что останется за рамками решения и указывая критерии, по которым заинтересованные лица будут судить о готовности решения. Рамки создаются на основе единого видения, являются результатом компромисса между сформулированными целями и условиями реальности и отражают приоритезацию заказчиком имеющихся требований к создаваемому решению. Частью процесса определения рамок проекта является вынесение менее важной функциональности из текущего проекта в планы на будущее.

*Рамки решения (solution scope)* определяют функциональность решения и его возможности (включая те, что не относятся к программному обеспечению). *Возможность* (*функциональность*, *составляющая*, *feature*) – это требуемый или желаемый аспект программного или аппаратного обеспечения. Например, предварительный просмотр перед печатью может быть возможностью текстового процессора; шифрование почтовых сообщений – возможностью почтовой программы. Сопроводительные руководства пользователей, интерактивные файлы помощи, операционные руководства и обучение также могут быть составляющими решения.

*Рамки проекта (project scope)* определяют объем работ, который должен быть выполнен проектной группой для поставки заказчику каждого из элементов, определенного рамками решения.

Управление рамками проекта критично для его успеха. Необходимо определять и фиксировать рамки решения и проекта, используя *треугольник компромиссов* и *матрицу компромиссов проекта*.

**Функциональность решения**

Регистрация и авторизация пользователей, хранение информации он них в БД.

Реализация личного кабинета пользователя в котором он может

* Посмотреть свои проекты / персонажей
* Удалить проект /персонажей
* Добавить проект/персонажей

Возможность сохранить проект в формате pdf

Данная функциональность будет реализована и в телеграм боте и на сайте, пункты ниже будут реализованы только на сайте

Изменить проект с помощью редактора, а именно

* Добавить страницу
* Выбрать шаблон ( разделение страницы на блоки )
* В блоках для текста он может ввести текст
* В блоках для изображения он может выбрать задний фон и персонажа

**За рамками решения**

* Возможность выбирать формат для сохранения проекта (png,jpg)
* Возможность редактирования текста (выбор стиля, размера, выравнивания)
* Добавление подсказок для пользователей

**Критерии одобрения решения**

* Вся функциональность корректно работает
* ИИ должна генерировать более 50 вариаций персонажа в разных позах и настроениях
* Количество шаблонов и фонов не менее 10
* Пользователь так же может скачать свою работу в формате pdf.

**Стратегии дизайна решения**

**Стратегия архитектурного дизайна**

На основе разработанного списка возможностей и функций формируется *стратегия архитектурного дизайна* (*architectural design strategy*), описывающая решение в целом. Она определяет компоненты решения и их взаимодействие. Отличный способ описания решения на этом этапе – использование иллюстрирующих диаграмм (например, UML).

Будет разработана Клиент-серверная архитектура с компонентами:

* React (фронтенд), - Клиент, отправляет запросы на сервер отображает интерфейс пользователю
* Django (бэкенд) - Сервер, принимает запросы с клиента и возвращает ответы
* PostgreSQL (БД) - Хранит информацию о пользователях
* midjourney (ИИ) - Создает персонажей на основе запросов или изображения

Добавить uml диаграмму

**Стратегия технологического дизайна**

Разработка решения потребует использования определенных продуктов и технологий. *Стратегия технологического дизайна* (*technical design strategy*) описывает, какие технологии и продукты выбраны проектной группой в качестве средства реализации решения.

**Фронтенд:** React

**Бэкенд:** Django

**БД:** PostgreSQL

**ИИ:** nano banana, midjourney

**Архитектурные паттерны:**

**MVVP** — разделяет логику приложения и интерфейса, обеспечивая чистоту кода и упрощение тестирования.

**Фабрика** — применяется для создания персонажей и проектов не расскрывая их внутренней реализации

**Объекты** — описание всех сущностей (пользователи, проекты, персонажи и шаблоны) как классов со своими полями и методами.

1. [↑](#footnote-ref-0)