МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

КУРСОВОЕ ПРОЕКТИРОВАНИЕ
Разработка прототипа веб-сервиса «МойРацион»
Б-ИН-22-01-186226.2470-а.09.000 КП

Студент
гр. БИН-22-02 _____ Е.К. Гордиенко

Руководитель,
канд. техн. наук, доцент _____ Е. Ю. Соболевская

Владивосток 2026

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)

Институт информационных технологий и анализа данных

Кафедра информационных технологий и систем

Индивидуальное задание
на производственную технологическую (проектно-технологическую) практику

Студенту гр. БИН-22-2 Гордиенко Елене Константиновне

Задание. Проанализировать и обосновать технологический стек разработки. Спроектировать архитектуру и пользовательский опыт. Разработать и протестировать интерактивный прототип.

Структура отчета по практике: титульный лист, оглавление, введение, основная часть, заключение, список использованных источников.

Отчет по практике оформляется в соответствии с СК-СТО-ТР-04-1.005-2015 «Требования к оформлению текстовой части выпускных квалификационных работ, курсовых работ (проектов), рефератов, контрольных работ, отчётов по практикам, лабораторным работам».

Срок сдачи отчета на кафедру: 14.01.2026

Руководитель,
кандидат техн. наук,
доцент _____ Соболевская Е. Ю.

Задание получил: _____ Гордиенко Е. К.

Аннотация

Курсовая работа посвящена подготовке к разработке прототипа веб-сервиса «МойРацион», предназначенного для автоматизации планирования сбалансированного питания. В работе последовательно решены задачи по обоснованию технологического стека, проектированию пользовательского интерфейса и проведению юзабилити-тестирования интерактивного прототипа.

На первом этапе выполнен сравнительный анализ платформ разработки, языков программирования, фреймворков и инструментов управления проектом. В результате обоснован выбор микросервисной архитектуры, стека технологий (Python/FastAPI для backend, React для frontend, PostgreSQL для базы данных, Docker для контейнеризации, GitLab для CI/CD) и облачной платформы Yandex Cloud.

Вторая часть работы охватывает полный цикл проектирования UX/UI: от разработки визуальной идентичности и пользовательских сценариев до создания wireframes, детальных макетов (Mockup) и интерактивного кликабельного прототипа в Figma.

Заключительный этап включает проведение юзабилити-тестирования прототипа с представителями целевой аудитории, по результатам которого выявлены и устранены ключевые проблемы взаимодействия.

Результатом работы является полностью подготовленный к реализации прототип веб-сервиса, подкреплённый аргументированным выбором технологий и проверенный на соответствие критериям удобства и эффективности.

Содержание

Введение

Разработка современных веб-сервисов представляет собой сложный процесс, требующий не только владения техническими навыками, но и строгого методологического подхода на всех этапах создания продукта – от концепции до реализации. Успешный стартап (далее по тексту «проект») основывается на тщательном выборе инструментария и глубоком понимании потребностей конечного пользователя.

Актуальность данной работы обусловлена растущей конкуренцией на рынке цифровых сервисов, где ключевыми факторами успеха становятся не только функциональность, но и качество пользовательского опыта, скорость разработки и стабильность итогового продукта. Особую важность приобретает начальная фаза проекта, на которой закладывается его архитектурный и интерфейсный фундамент.

Целью работы является обоснование технологического стека и проектирование пользовательского интерфейса для веб-сервиса «МойРацион». Для достижения поставленной цели необходимо решить ряд последовательных задач.

Во-первых, провести сравнительный анализ и обосновать выбор программного обеспечения для разработки, включая платформы, языки программирования, фреймворки, библиотеки и инструменты для управления проектом.

Во-вторых, системно подойти к прототипированию интерфейса, пройдя полный цикл от разработки UI/UX-концепции и пользовательских сценариев до создания визуальных макетов (Mockup) и интерактивного кликабельного прототипа.

В-третьих, валидировать проектные решения через юзабилити-тестирование, обеспечив соответствие создаваемого продукта ожиданиям целевой аудитории.

Данный отчет структурирован в соответствии с логикой разработки. Первая глава посвящена анализу и выбору технологического стека. Вторая глава описывает процесс прототипирования, начиная с дизайна и заканчивая тестированием интерактивной модели. Результатом работы станет полностью готовый к реализации прототип, подкрепленный аргументированным выбором инструментов и проверенный на соответствие критериям удобства и эффективности.

1 Выбор программного обеспечения для разработки

1.1 Анализ платформ разработки

Разработка веб-сервиса «МойРацион» требует тщательного выбора программного обеспечения, которое обеспечит реализацию функциональных и нефункциональных требований, описанных в техническом задании. Ключевыми критериями при выборе являются: поддержка глубокой персонализации, возможность интеграции с внешними сервисами, высокая производительность, безопасность данных, масштабируемость и удобство разработки.

Для реализации веб-сервиса «МойРацион» рассматривались две основные архитектурные платформы: монолитная и микросервисная (таблица 1). Монолитная архитектура предполагает разработку единого приложения, где все компоненты (фронтенд, бэкенд, база данных) тесно связаны. Микросервисная архитектура, в свою очередь, разделяет приложение на независимые сервисы, каждый из которых отвечает за определённую функциональность. Для наглядности и обоснованности последующего выбора проведён сравнительный анализ по наиболее значимым для проекта критериям.

Таблица 1 – Сравнительный анализ архитектур для веб-сервиса «МойРацион»

Критерий	Монолитная архитектура	Микросервисная архитектура
Масштабируемость	Вертикальное масштабирование всего приложения целиком. Сложно масштабировать отдельные компоненты.	Горизонтальное масштабирование независимых сервисов. Можно усилить только нагруженные модули.
Надёжность и доступность	Единая точка отказа. Проблема в одном модуле может «положить» всю систему.	Отказ одного сервиса не приводит к остановке других. Повышает отказоустойчивость и доступность.
Гибкость и обновления	Любое обновление требует пересборки и перезапуска всего приложения. Медленные итерации.	Каждый сервис обновляется и развёртывается независимо. Позволяет быстро вносить изменения.
Технологический стек	Единый стек для всего приложения. Сложно применять разные инструменты под разные задачи.	Разные сервисы могут использовать оптимальные для их задач языки и фреймворки.

Продолжение таблицы 1

Критерий	Монолитная архитектура	Микросервисная архитектура
Интеграции	Все интеграции внутри одной кодовой базы. Риск конфликтов и сложных зависимостей.	Внешние сервисы оборачиваются в отдельные микросервисы. Чёткие границы и изоляция.
Безопасность	Единая поверхность атаки. Уязвимость в одном компоненте угрожает всей системе.	Распределённая поверхность атаки. Изоляция сервисов ограничивает потенциальный ущерб.

На основе проведённого анализа для веб-сервиса «МойРацион» выбрана микросервисная архитектура. Этот выбор обусловлен её принципиальным соответствием стратегическим целям проекта – созданию масштабируемой, высокодоступной и гибкой цифровой платформы. Монолитная архитектура, будучи проще на старте, не обеспечивает необходимой степени отказоустойчивости и независимого развития ключевых модулей, таких как интеллектуальный планировщик рационов, система интеграции с партнёрами и образовательный компонент. Микросервисный подход позволяет не только эффективно распределять нагрузку в условиях роста пользовательской базы, но и обеспечивает изоляцию сервисов, что критически важно для безопасности персональных данных и бесперебойной работы сервиса в целом. Учитывая требования к масштабируемости (поддержка роста пользовательской базы), интеграции с внешними сервисами (доставка, картография) и необходимость обеспечения высокой доступности (99,5%), выбрана микросервисная архитектура. Она позволит независимо масштабировать модули, такие как система планирования питания, мониторинг пищевого поведения, интеграционный и образовательный модули. Это также упростит внедрение обновлений и снизит риски сбоев всей системы.

Для хостинга предполагается использование облачных провайдеров, что соответствует требованиям к надежности и географической доступности для пользователей из регионов России. Облачный провайдер – это поставщик услуг на основе облачных вычислений с использованием виртуальных мощностей, таких как серверы, системы хранения данных. Благодаря облачным технологиям компаниям не нужно закупать дорогостоящее оборудование, а также нанимать сотрудников для настройки и обслуживания [1]. В качестве платформы для развертывания микросервисов рассматриваются Docker. Docker обеспечит контейнеризацию каждого сервиса.

На этапе контейнеризации микросервисов необходимо выбрать технологию, которая обеспечит изоляцию, переносимость и управляемость каждого компонента. Основное сравнение проводится между двумя современными решениями – Docker (фактический стандарт рынка) и Podman (более новая альтернатива с иной архитектурой). Docker – это платформа с

открытым исходным кодом для автоматизации разработки, доставки и развертывания приложений. Ее основная идея – создание стандартного и предсказуемого окружения, где приложения могут работать независимо от операционной системы или инфраструктуры [2]. Podman – платформа для управления контейнерами, позволяющая виртуализировать приложения/процессы [3]. Также для полноты картины рассматривается вариант использования виртуальных машин (ВМ), как более традиционный, но всё ещё применяемый подход. Сравнительный анализ по ключевым для проекта критериям представлен в таблице 2.

Сравнение с Podman и виртуальными машинами подтверждает, что Docker предлагает оптимальный баланс. От виртуальных машин он выгодно отличается лёгкостью, скоростью и идеальной интеграцией с Kubernetes, что критично для микросервисной архитектуры. Хотя Podman предлагает более безопасную архитектуру, решающим фактором в пользу Docker стала его беспрецедентная зрелость экосистемы, статус отраслевого стандарта и, как следствие, минимальные риски при интеграции, развёртывании и поиске готовых решений, что ускоряет вывод продукта на рынок.

Таблица 2 – Сравнение инструментов изоляции приложений

Критерий	Docker	Podman	Виртуальные машины
Производительность	Высокая (лёгкие контейнеры)	Высокая (лёгкие контейнеры)	Низкая (тяжёлые изолированные ОС)
Экосистема	Огромная, стандарт индустрии	Меньше, но совместима с Docker	Зрелая, но для другой задачи
Безопасность	Средняя (работа через демона)	Выше (работа без демона)	Максимальная (полная изоляция)
Скорость развёртывания	Секунды	Секунды	Минуты

Выбор облачной платформы является стратегическим решением, влияющим на производительность, соответствие законодательству и стоимость владения. Основное сравнение проводится между международным гигантом Amazon Web Services (AWS) и локальным лидером Yandex Cloud, так как они представляют два принципиально разных подхода для российского рынка. Для полноты анализа в качестве третьего варианта рассмотрена аренда bare-metal серверов (VDS/VPS) у хостинг-провайдера, что представляет собой традиционный способ размещения инфраструктуры. Сравнение по ключевым критериям представлено в таблице 3.

Таблица 3 – Сравнение вариантов инфраструктуры

Критерий	Yandex Cloud	Amazon AWS	Обычный хостинг (VPS)
Скорость в России	Максимальная	Средняя (серверы в Европе)	Зависит от провайдера
Управляемый Kubernetes	Есть (Yandex Managed Kubernetes)	Есть (Amazon EKS)	Нет, нужно настраивать самим

Продолжение таблицы 3

Критерий	Yandex Cloud	Amazon AWS	Обычный хостинг (VPS)
Соответствие 152-ФЗ	Полное, «из коробки»	Требует дополнительных действий	Нужно проверять отдельно
Масштабируемость	Автоматическая	Автоматическая	Ручная, сложная
Стоимость	Прозрачная, в рублях	Сложная, в валюте	Фиксированная, но нужен администратор
Интеграция с сервисами	Лучшая для Яндекса (Карты, Доставка)	Через API	Через API

В качестве облачного провайдера для проекта выбран Yandex Cloud. Сравнение трёх вариантов чётко демонстрирует его целесообразность. Использование bare-metal/VPS отпадает сразу, так как требует огромных трудозатрат на построение и поддержку отказоустойчивого Kubernetes-кластера, что противоречит принципам эффективности и масштабируемости. Между AWS и Yandex Cloud выбор сделан в пользу последнего по совокупности критически важных для «МойРацион» факторов: минимальная задержка для пользователей в России, гарантированное соответствие 152-ФЗ «О персональных данных» и нативная, глубокая интеграция с экосистемой Яндекса (Карты, Доставка), которая является ключевой для функционала сервиса. Это обеспечивает не только техническую и правовую основу, но и значительное конкурентное преимущество на целевом рынке.

1.2 Выбор языка программирования

Выбор языка программирования для backend-разработки обусловлен необходимостью реализации сложных алгоритмов персонализации, работы с большими объемами данных, высокой производительности и безопасности. Для корректного сравнения рассмотрены три ключевых варианта, представляющих разные парадигмы разработки. Сравнительный анализ представлен в таблице 4. Python как лидер в области науки о данных, Node.js для высоконагруженных сетевых приложений и Java как классический выбор для корпоративных систем.

Таблица 4 – Сравнение языков программирования для backend-разработки

Критерий	Python	Node.js	Java
Сложные алгоритмы и ML	Лучшие возможности (Pandas, NumPy, Scikit-learn)	Ограниченная поддержка (библиотеки менее зрелые)	Хорошие возможности (Weka, DL4J), но сложнее в реализации
Скорость разработки	Максимальная	Высокая (единый язык с фронтендом)	Ниже (строгая типизация, более объемный код)

Продолжение таблицы 4

Критерий	Python	Node.js	Java
Безопасность	Хорошая (зависит от фреймворка и практик)	Хорошая (зависит от фреймворка)	Отличная (строгая типизация, проверки на этапе компиляции)
Интеграции с API	Отличная (широкие возможности, библиотеки requests, FastAPI)	Отличная (нативная работа с JSON, асинхронные запросы)	Хорошая (требует больше кода для сериализации)
Масштабируемость	Средняя (GIL ограничивает многопоточность)	Высокая (event-loop, микросервисы)	Высокая (многопоточность, кластеризация)
Сообщество и библиотеки	Огромное в data science и вебе	Огромное в веб-разработке	Огромное
Производительность	Средняя (интерпретируемый язык)	Высокая (движок V8, асинхронность)	Высокая (JIT-компиляция)

Для backend-разработки «МойРацион» выбран Python в связке с фреймворком FastAPI. Python – это высокоуровневый язык программирования, отличающийся эффективностью, простотой и универсальностью использования [4]. Анализ показывает, что Python обеспечивает наиболее сбалансированное соответствие ключевым требованиям проекта. Хотя Node.js предлагает высокую производительность для операций ввода-вывода, он существенно уступает в возможностях реализации сложных алгоритмов персонализации и машинного обучения, которые составляют ядро ценностного предложения сервиса. Java обеспечивает высокую надежность и производительность, но требует значительно больше времени на разработку и менее удобен для быстрого прототипирования, что критично для стартапа на этапе валидации гипотез. Python позволяет быстро реализовать интеллектуальные модули планирования питания благодаря богатейшим библиотекам, обеспечивает удобную интеграцию с внешними API и соответствует требованиям к скорости разработки, указанным в дорожной карте, сохраняя при этом достаточный уровень производительности для прогнозируемых нагрузок.

Для frontend-разработки основное сравнение проводится между тремя современными фреймворками: React как наиболее популярное решение с виртуальным DOM, Vue.js с его прогрессивной архитектурой и Angular как полноценный фреймворк для крупных корпоративных приложений. Сравнение представлено в таблице 5.

Для frontend-разработки «МойРацион» выбран JavaScript с использованием фреймворка React. Этот выбор обусловлен тем, что React обеспечивает создание интуитивно понятного, адаптивного и производительного интерфейса, что полностью соответствует пользовательским требованиям к удобству использования.

Таблица 5 – Сравнение фреймворков для frontend-разработки

Критерий	React	Vue.js	Angular
Кривая обучения	Средняя (нужно знать JSX, хуки)	Низкая (постепенное внедрение)	Высокая (TypeScript, сложные концепции)
Гибкость	Максимальная (только библиотека UI)	Высокая (прогрессивный фреймворк)	Ограниченная (полноценный фреймворк)
Производительность	Высокая (виртуальный DOM)	Высокая	Высокая
Мобильная разработка	React Native (кроссплатформенно)	Vue Native (менее развит)	Ionic (кроссплатформенно)
Интеграция с бэкендом	Отличная (независим от бэкенда)	Отличная	Отличная
Типизация	Опциональная (через TypeScript)	Опциональная	Обязательная (TypeScript)
Сообщество и рынок	Крупнейшее, большинство вакансий	Большое и быстро растущее	Крупное, корпоративный сегмент

По сравнению с Vue.js, React имеет более зрелую экосистему и лучше подходит для сложных динамических интерфейсов с большим количеством интерактивных элементов, которые характерны для персонализированного дашборда сервиса. Angular, несмотря на свою мощь и строгую архитектуру, избыточен для текущего этапа проекта и создает ненужную сложность для команды. Выбор React также открывает стратегические возможности для будущего развития продукта, позволяя использовать React Native для создания кроссплатформенного мобильного приложения с сохранением единой логики компонентов и компетенций команды.

1.3 Выбор фреймворков и библиотек

Выбор конкретных инструментов разработки требует тщательного анализа, поскольку определяет не только скорость реализации, но и архитектурные возможности, производительность и долгосрочную сопровождаемость системы. Каждый компонент стека оценивался с точки зрения его соответствия функциональным требованиям «МойРацион», таким как необходимость обработки сложных данных для персонализации, обеспечение отказоустойчивости интеграций и создание отзывчивого пользовательского интерфейса.

В предыдущем подразделе, уже определены нужные фреймворки для frontend и backend частей проекта. Для frontend-части выбран JavaScript с использованием фреймворка React. Для backend-разработки «МойРацион» выбран Python в связке с фреймворком FastAPI. Однако важно рассмотреть проект глубже, понять какие еще фреймворки и библиотеки для проекта необходимы.

Фоновые задачи, такие как генерация индивидуальных планов питания и синхронизация с внешними API, требуют надёжной и масштабируемой системы очередей. Celery представляет собой промышленное решение с поддержкой распределённых воркеров, планировщиком периодических задач и продвинутыми механизмами мониторинга. Его архитектура позволяет гибко масштабировать обработку задач, что соответствует прогнозам роста пользовательской базы. RQ предлагает более простую модель, которая, хотя и проще в настройке, не предоставляет встроенных инструментов для сложного планирования и мониторинга, что может стать ограничением при увеличении количества асинхронных операций. Выбор Celery продиктован требованиями к надёжности и управляемости процессов, которые напрямую влияют на пользовательский опыт – задержка в генерации плана питания или сбои при интеграции с доставкой недопустимы.

Выбор системы хранения данных является фундаментальным, поскольку определяет возможности аналитики, целостность информации и масштабируемость. Для хранения данных рассматривались PostgreSQL, MySQL и MongoDB как представители разных подходов к хранению данных (таблица 6).

Таблица 6 – Сравнение систем управления базами данных

Критерий	PostgreSQL	MySQL	MongoDB
Тип данных	Реляционная (SQL)	Реляционная (SQL)	Документная (NoSQL)
Транзакции	Полная поддержка ACID	Поддержка ACID	Ограниченная
Сложные запросы	Отличные (оконные функции, CTE)	Хорошие	Ограниченные
Масштабируемость	Вертикальная и горизонтальная	Вертикальная	Горизонтальная
Геопространственные данные	Отличная поддержка (PostGIS)	Ограниченная	Хорошая
Соответствие требованиям	Полное	Высокое	Частичное

Выбрана PostgreSQL как наиболее полнофункциональная реляционная СУБД. Её поддержка сложных запросов, оконных функций и транзакций ACID критически важна для корректного учёта пищевого поведения пользователей и формирования аналитических отчётов. Встроенная поддержка геопространственных данных через PostGIS также обеспечивает основу для будущего развития картографического модуля.

1.4 Выбор инструментов для прототипирования

Процесс проектирования интерфейса является критически важным этапом, напрямую влияющим на итоговый пользовательский опыт и успех продукта. Выбор инструментов для

этой задачи должен обеспечивать не только визуализацию идей, но и эффективную коммуникацию между дизайнерами, разработчиками и продуктовыми аналитиками, а также позволять проводить быстрые итерации на основе обратной связи.

Для веб-сервиса «МойРацион» требовался инструмент, который поддерживал бы весь цикл дизайн-разработки: от исследования пользователей и построения карты путешествия до создания интерактивных прототипов, и передачи готовых макетов в разработку.

Проведённый анализ рынка инструментов проектирования выявил три ключевых решения, каждое из которых занимает свою нишу (таблица 7). Figma позиционируется как облачная платформа для совместной работы над дизайном в реальном времени, ставшая отраслевым стандартом. Adobe XD предлагает глубокую интеграцию с экосистемой Adobe и мощные возможности прототипирования. Sketch, который пионером в области цифрового дизайна интерфейсов, остаётся популярным выбором, особенно среди пользователей macOS, но его локальная природа и зависимость от плагинов для совместной работы создают определённые ограничения.

Таблица 7 – Сравнение инструментов проектирования UI/UX

Критерий	Figma	Adobe XD	Sketch
Платформа и доступность	Кроссплатформенный веб-инструмент, работает в браузере	Кроссплатформенное нативное приложение	Только macOS, требуется установка
Совместная работа в реальном времени	Полноценная, несколько пользователей могут работать в одном файле одновременно	Ограниченная (копросмотр), без полноценного одновременного редактирования	Через сторонние плагины и сервисы, с задержками
Прототипирование и интерактивность	Мощные возможности для создания сложных переходов и логики внутри платформы	Глубокие возможности, тесная интеграция с After Effects для анимаций	Базовые возможности, часто требуются сторонние инструменты
Система компонентов и дизайн-система	Автоматические компоненты, мощные возможности для создания библиотек	Компоненты и состояния, удобное управление активами	Символы и библиотеки, но менее гибкие
Интеграция с процессами разработки	Прямая генерация CSS-кода, экспорт ресурсов, плагины для Jira, Slack	Интеграция с Creative Cloud, плагины для разработчиков	Зависит от плагинов (Zeplin, Avocode) для передачи макетов
Модель распространения и стоимость	Freemium модель, бесплатный стартовый план с ограничениями	Часть подписки Adobe Creative Cloud	Разовая покупка лицензии на Mac

Для проектирования пользовательского интерфейса и создания кликабельных прототипов, соответствующих дорожной карте, выбран инструмент Figma. Ключевым фактором этого выбора стала его облачная природа и превосходные возможности для командной работы.

В отличие от локальных решений, Figma позволяет дизайнеру, продуктовому аналитику и разработчику одновременно просматривать и комментировать макет в реальном времени, что критически важно для распределённой команды, работающей над согласованием сложных пользовательских сценариев сервиса «МойРацион». Возможность создавать адаптивные макеты с помощью Auto Layout и Constraints напрямую отвечает требованию поддержки различных устройств – от смартфонов до десктопов.

Встроенные инструменты прототипирования позволяют быстро «оживить» статические экраны, соединив их в интерактивные потоки для проверки гипотез и проведения юзабилити-тестирования с пользователями без написания кода. Кроме того, Figma упрощает процесс передачи дизайна в разработку: разработчики могут инспектировать макеты, получать точные CSS-стили и экспортировать ресурсы непосредственно из интерфейса, что сокращает количество ошибок и ускоряет реализацию.

Помимо инструмента для детального дизайна интерфейсов, на этапе стратегического проектирования необходим инструмент для визуализации и структурирования пользовательских исследований. Здесь основными альтернативами выступили Miro как лидер рынка цифровых досок и MURAL как его основной конкурент, предлагающий схожий функционал (таблица 8).

Таблица 8 – Сравнение инструментов для визуализации исследований и стратегии

Критерий	Miro	MURAL
Интеграция с другими инструментами	широкая интеграция (Jira, Confluence, Slack, Figma)	глубокая интеграция с Microsoft Teams, Zoom
Гибкость и шаблоны	огромная библиотека шаблонов для CJM, User Story Mapping, мозговых штурмов	фокус на шаблонах для дизайн-спринтов и agile-практик
Совместная работа	неограниченное количество участников на доске в реальном времени	аналогичные возможности, но с акцентом на фасилитацию сессий
Простота освоения	интуитивный интерфейс, низкий порог входа	немного более сложный интерфейс, ориентированный на профессионалов

Для визуализации пользовательских сценариев, построения карт путешествия, выбран Miro. Этот выбор обусловлен его неограниченной гибкостью, обширной библиотекой шаблонов, идеально подходящих для методов, и беспрепятственной интеграцией с Figma.

Использование Miro позволяет превратить абстрактные результаты исследований в наглядные схемы и диаграммы, которые становятся единым источником истины для всей команды и обеспечивают согласованность в понимании потребностей целевой аудитории на всех этапах разработки. Комбинированное использование Figma и Miro создаёт полноценную экосистему проектирования, охватывающую как стратегический, так и тактический уровень создания продукта.

1.5 Система управления версиями

Система управления версиями – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое [5].

Выбор системы управления версиями и платформы для хостинга репозиториев является фундаментальным решением, которое закладывает основу для всего процесса разработки, влияя на скорость внедрения изменений, качество кода, безопасность и возможность командной работы.

Для проекта «МойРацион», предполагающего микросервисную архитектуру с распределённой командой, критически важно обеспечить строгий контроль над изменениями в каждом компоненте, автоматизировать процессы сборки и тестирования, а также иметь прозрачный механизм отслеживания задач и ошибок.

Исходный код представляет собой одну из ключевых ценностей проекта, и его защита, управление доступом и документация изменений прямо связаны с требованиями к информационной безопасности и сопровождению.

Платформа для хостинга Git-репозиториев должна не только предоставлять надёжное хранилище, но и интегрировать в единую среду инструменты для непрерывной, управления задачами, ревью кода и мониторинга. На рынке доминируют три основных решения, каждое со своей философией и экосистемой (таблица 9).

GitLab позиционируется как единая платформа DevOps, предлагающая полный цикл инструментов «из коробки». GitHub, приобретённый Microsoft, является крупнейшим сообществом разработчиков с акцентом на социальное взаимодействие и обширную экосистему сторонних интеграций. Bitbucket от Atlassian традиционно силён в интеграции с инструментами для управления проектами, такими как Jira и Confluence, что делает его популярным в корпоративной среде.

Для контроля версий кода и организации командной работы в проекте «МойРацион» выбрана система Git с использованием платформы GitLab. Git – это специальная программа,

которая позволяет отслеживать любые изменения в файлах, хранить их версии и оперативно возвращаться в любое сохранённое состояние. Большинство других систем контроля версий хранят информацию в виде списка изменений в файлах.

Git работает иначе – он хранит скорее набор снимков – полное отображение того, как выглядит файл в момент сохранения. Это позволяет всегда иметь полную информацию обо всех файлах и быстро восстанавливать любую из предыдущих версий [6].

Таблица 9 – Сравнение платформ для хостинга Git-репозиториев

Критерий	GitLab	GitHub	Bitbucket
Модель распространения и стоимость	Бесплатный мощный тариф для частных репозиториев, self-hosted Community Edition.	Бесплатные приватные репозитории, платный корпоративный функционал.	Бесплатный тариф с ограничениями, глубокая интеграция с Atlassian.
Инструменты CI/CD	Встроенный, мощный GitLab CI/CD с детальной конфигурацией в gitlab-ci.yml.	GitHub Actions (гибкие workflow через YAML).	Bitbucket Pipelines (интеграция с Docker, ограниченные минуты на бесплатном тарифе).
Управление проектами и issues	Встроенный трекер задач, доски Kanban, планирование Milestones.	Issues, Projects (доски), интеграция с внешними трекерами.	Глубокая нативная интеграция с Jira, собственный трекер задач.
Безопасность и контроль доступа	Детальные роли доступа, защита веток, security scanning (SAST, DAST).	Организации, teams, защищённые ветки, Dependabot для уязвимостей.	Гибкое управление доступом, интеграция с корпоративными SSO.
Интеграции и экосистема	Большой Marketplace с интеграциями, но меньше, чем у GitHub.	Крупнейшая экосистема приложений и действий.	Фокус на интеграции с CI/CD-инструментами.

Данный выбор обусловлен принципиальным подходом GitLab, который предлагает полноценную DevOps-платформу в едином интерфейсе.

В отличие от конкурентов, где CI/CD-инструменты (GitHub Actions, Bitbucket Pipelines) являются дополнением, в GitLab конвейер сборки, тестирования и развёртывания является неотъемлемой и тесно интегрированной частью системы. Это идеально соответствует требованиям микросервисной архитектуры, где необходимо настраивать отдельные, но скоординированные пайплайны для десятков сервисов.

Наличие встроенного Container Registry для Docker-образов и удобных инструментов для развёртывания в Kubernetes (Kubernetes integration, Auto DevOps) значительно упростит и автоматизирует процессы, критичные для соблюдения сроков по дорожной карте.

Важным аспектом выбора GitLab стала его гибкая модель лицензирования, предоставляющая мощный бесплатный тариф для приватных репозиториев и всей команды. Это сни-

жает начальные операционные расходы на старте проекта. Кроме того, встроенные возможности статического и динамического анализа безопасности кода напрямую отвечают строгим требованиям к защите персональных данных пользователей.

Детальный контроль доступа на уровне веток, система мердж-реквестов с обязательным ревью кода и ведение полной, неизменяемой истории изменений обеспечивают необходимый уровень документации и соответствие стандартам разработки, что особенно важно для проекта в области цифрового здоровья.

Использование единой платформы от хранения кода до его доставки в продакшен позволяет минимизировать количество используемых инструментов, снизить накладные расходы на их интеграцию и сконцентрировать усилия команды на разработке продукта, а не поддержке инфраструктуры.

2 Прототипирование веб-сервиса «МойРацион»

2.1 Визуальная идентичность веб-сервиса

Аббревиатура UX расшифровывается как user experience – «пользовательский опыт» – это то, каким образом пользователь взаимодействует с интерфейсом и насколько сайт или приложение для него удобны. В UX входит навигация по сайту, функционал меню и результат взаимодействия со страницами. Именно от качества UX зависит то, насколько быстро пользователь сможет получить то, зачем он пришёл на сайт.

UI – это user interface, пользовательский интерфейс – оформление сайта: сочетания цветов, шрифты, иконки и кнопки. В современном дизайне UX и UI практически всегда идут рядом, потому что они очень тесно связаны [7].

UI/UX дизайн веб-сервиса «МойРацион» формировался как прямой ответ на ключевые инсайты, полученные в ходе исследования целевой аудитории. Анализ показал, что пользователи разделены на три сегмента, каждый со своими приоритетами: занятые профессионалы ценят скорость и автоматизацию, начинающие энтузиасты нуждаются в наглядности и мотивации, а целевые приверженцы ЗОЖ требуют глубокой кастомизации и точности данных. Задача дизайна заключалась в том, чтобы создать единый, адаптивный интерфейс, который персонализируется под каждого пользователя, делая сложные процессы простыми и интуитивно понятными.

Цветовая схема – это набор нескольких оттенков, которые будут использоваться на сайте [8]. Разработка цветовой схемы велась с учётом психологического восприятия и тематики здоровья. Оттенки выбраны с учетом возрастов целевой аудитории, по результатам анкетирования которая оказалась в возрастом диапазоне 18-45 лет. Основным акцентным цветом выбран зелёный (#8CC544), который ассоциируется со свежестью, здоровьем и гармонией. Этот цвет применяется для ключевых призывов к действию и позитивных индикаторов выполнения нормы. Для фона и нейтральных элементов использована гамма тёплых серых оттенков, создающая спокойную и сосредоточенную атмосферу, которая не отвлекает от контента (#F1F1F1). Для выделения предупреждений и зон внимания выбран мягкий красный (#FF5E6). Такая палитра не только создаёт визуальную иерархию, но и соответствует принципам доступности, обеспечивая достаточный контраст для комфортного восприятия. Подробнее ознакомиться с палитрой можно на рисунке 1.

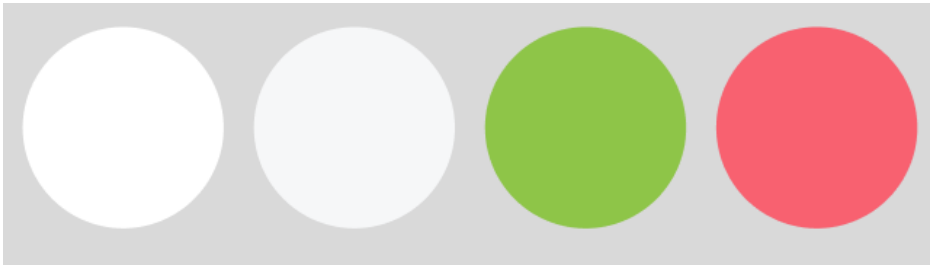

Рисунок 1 – Цветовая палитра

Типографика подобрана для обеспечения максимальной читаемости и отражения тонкости и чистоты. В качестве основной гарнитуры для интерфейса выбран NanumMyeongjo – чистый, современный и хорошо читаемый шрифт с множеством начертаний. Его геометрическая нейтральность позволяет легко выстраивать информационную иерархию. Для заголовков и акцентов используется более выразительный Monotype Corsiva, который добавляет динамики и индивидуальности.

Общий стиль интерфейса можно охарактеризовать как «минимализм». Дизайн очищен от излишней декоративности, чтобы пользователь мог сфокусироваться на данных и действиях. Ключевые информационные блоки оформлены в виде карточек, что создаёт ощущение упорядоченности и снижает визуальный шум.

Логотип сайта – это ключевой элемент визуальной идентификации бренда, играющий роль в формировании первого впечатления о компании, что крайне важно на современном рынке [9]. Независимо от его формы, цвета или стиля, правильные размеры и удачное размещение могут значительно повысить узнаваемость и улучшить восприятие пользователями. Разработка логотипа и графических элементов велась с акцентом на простоту и смысловую нагрузку. Логотип представляет собой стилизованное изображение яблока, перетекающего в вилку. Эта метафора объединяет идею здорового питания (яблоко) с повседневным процессом приема пищи (вилка), что отражает суть сервиса – гармоничное внедрение полезных привычек в ежедневную жизнь. Логотип выполнен в основной зелёной гамме и работает как в цветном, так и в монохромном вариантах. С логотипом можно ознакомиться на рисунке 2.

Рисунок 2 – Логотип веб-сервиса

Для сервиса также подобран набор дружелюбных иллюстраций в пастельных тонах, которые помогают снизить тревожность при знакомстве с темой питания и создают эмоциональную связь с пользователем. Пример иллюстрации отображен на рисунке 3.

Дизайн-система создавалась в Figma, что обеспечило единый источник истины для всех компонентов интерфейса (кнопки, формы, карточки, типографика) и ускорило работу команды. Каждый аспект визуального дизайна «МойРацион» продуман для того, чтобы превратить управление питанием из рутинной обязанности в осознанный, приятный и эффективный процесс.

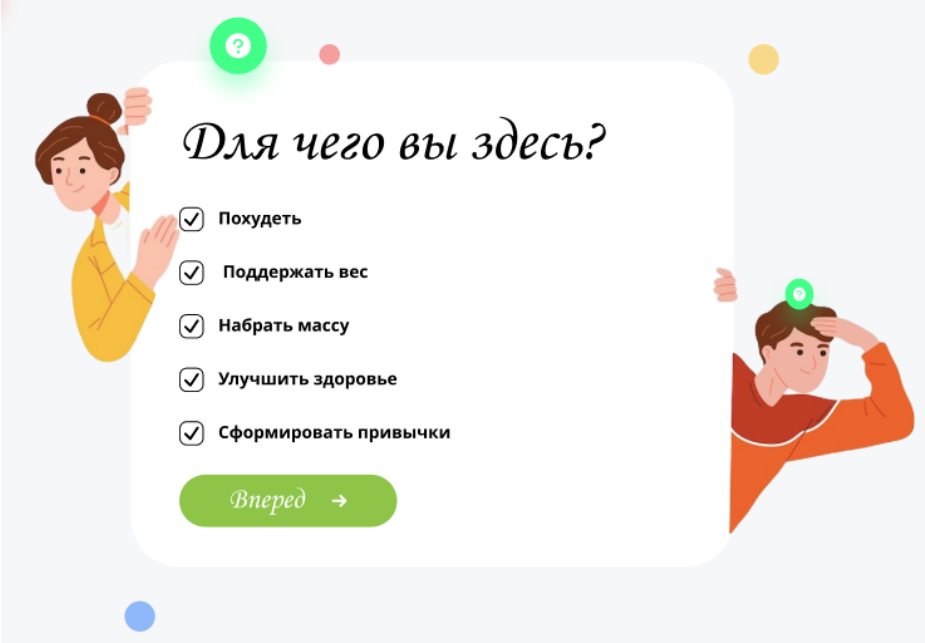

Рисунок 3 – Пример иллюстрации в веб-сервисе

Интерфейс становится не просто инструментом, а персональным проводником в мире здорового питания, что напрямую отвечает потребностям, выявленным в ходе анкетирования целевой аудитории.

2.2 Пользовательские сценарии

User Flow Map (карта пользовательского сценария) представляет собой визуализированную схему, которая отображает полный путь взаимодействия пользователя с цифровым продуктом. Этот инструмент позволяет проектировщикам и разработчикам понять логику использования сервиса, выявить потенциальные точки трения и оптимизировать пользовательский опыт [10].

Для веб-сервиса «МойРацион» разработана детальная карта пользовательского сценария, которая учитывает специфику работы с персонализированным питанием. Для лучшего понимания у карты есть легенда, которая отображена на рисунке 4.

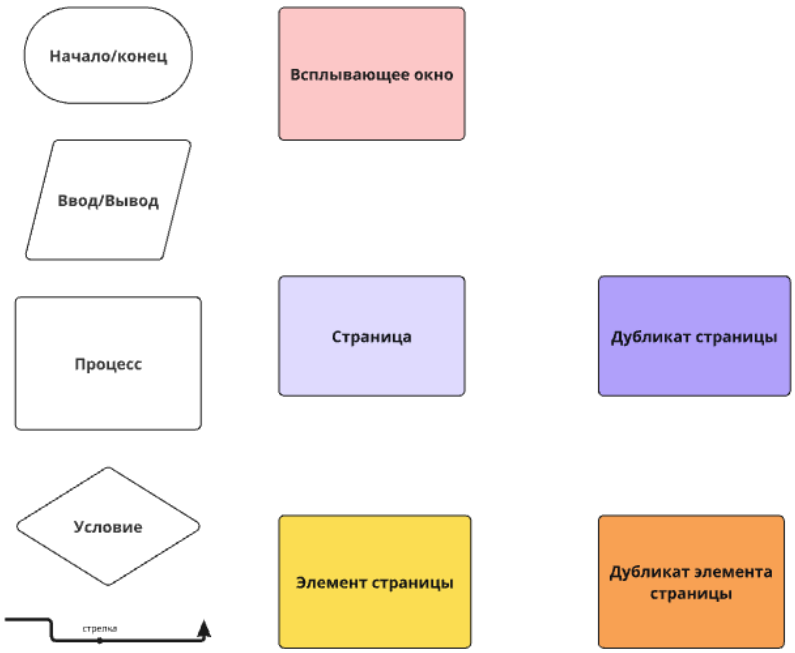

Рисунок 4 – Легенда User Flow Map

Представленная схема демонстрирует комплексный путь пользователя через все ключевые функциональные модули сервиса, ознакомиться с картой можно отсканировав кью-аркод в приложении А. Начальная точка взаимодействия предполагает знакомство пользователя с сервисом, после чего следует переход на страницу анкеты пользователя. Этап заполнения анкеты собирает персональные данные, необходимые для построения индивидуальной программы питания: антропометрические показатели, цели, пищевые предпочтения и ограничения.

После прохождения анкеты пользователь попадает на главную панель управления, которая служит центральным узлом для взаимодействия со всеми функциями сервиса. Отсюда возможен переход к персональному плану питания, который система формирует на основе введённых данных. Особое внимание на схеме уделено процессу корректировки плана: пользователь может инициировать пересчёт параметров питания или выполнить замену отдельных блюд и продуктов после процесса регистрации. Для этого предусмотрен механизм поиска альтернатив с применением фильтров по различным критериям, что обеспечивает гибкость и адаптивность системы. Эти этапы пути пользователя отражены на рисунке 5.

Рисунок 5 – Начальные этапы пути пользователя

Важной составляющей пользовательского сценария является блок дневник питания. Пользователь может вносить данные о фактически потреблённых продуктах через форму учёта питания, где система автоматически рассчитывает пищевую ценность и корректирует баланс нутриентов. Отдельно выделен функционал контроля водного баланса с возможностью отметки о выпитой жидкости, а также учёта физической активности. Это создаёт целостную картину пищевого поведения и способствует формированию осознанных привычек. Данные функции также доступны после прохождения этапов входа или регистрации. Пример этапа входа или регистрации можно рассмотреть на рисунке 6. Вход или регистрации происходят с помощью сторонних сервисов.

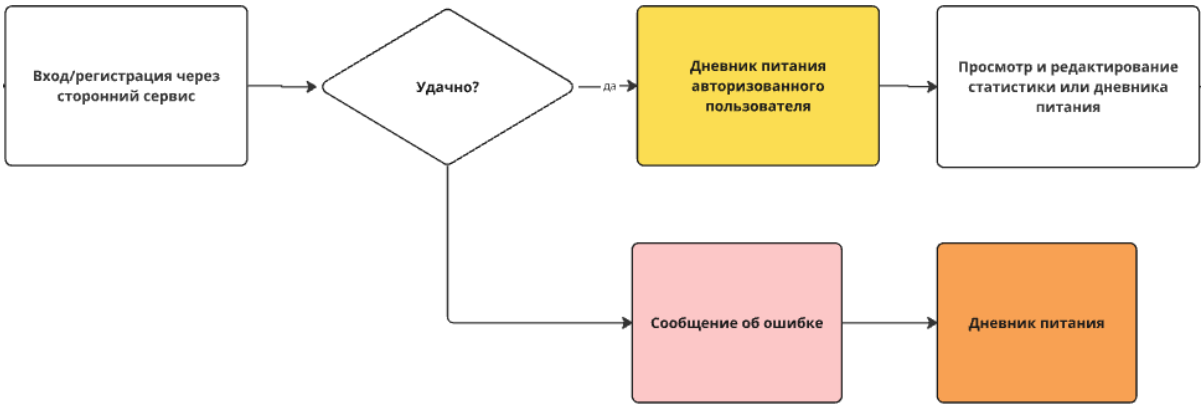

Рисунок 6 – Этап входа или регистрации пользователя

Образовательный компонент сервиса представлен в виде слайдера со статьями. Пользователь может изучать материалы по диетологии и нутрициологии, что способствует повышению пищевой грамотности и укреплению мотивации. Управление подпиской и настройками профиля выделено в отдельную ветку сценария, где пользователь может корректировать свои цели, обновлять данные и контролировать статус премиум-доступа.

Интеграционная составляющая сервиса отражена в блоке работы со списком покупок. Система автоматически формирует перечень необходимых продуктов на основе плана питания, а пользователь может искать магазины-партнёры через картографический модуль. Процесс оформления заказа включает проверку наличия товаров и подтверждение операции, что обеспечивает практическую реализацию сформированного рациона.

Карта пользовательского сценария демонстрирует, что веб-сервис «МойРацион» обеспечивает сквозной цикл взаимодействия: от первичной настройки и получения плана питания

до ежедневного мониторинга, обучения и практической реализации через интеграцию с сервисами заказа продуктов. Все элементы сценария взаимосвязаны и направлены на решение ключевых проблем пользователя: отсутствия времени на планирование, сложности самостоятельных расчётов и необходимости персонализации под индивидуальные особенности. Такая архитектура пользовательского опыта соответствует заявленному ценностному предложению сервиса и способствует достижению его основной миссии – формированию и поддержанию сбалансированного рациона питания через удобное и технологичное решение.

2.3 Wireframes

Вайрфрейм – это схема с низким уровнем детализации, которая визуализирует структуру и содержание цифрового проекта. Вайрфрейм показывает, как будут расположены все основные элементы продукта: навигация, карточки, текстовые блоки, иллюстрации и кнопки. В дизайне вайрфрейм – это скелет информационной архитектуры будущего продукта, который содержит все его значимые элементы в обобщённом виде. Цель вайрфрейма – показать общую картину [11]. Схему рисуют в двух-трёх цветах, обычно это серый, чёрный и белый. Вайрфреймы можно разделить на две группы: низкодетализированные и высокодетализированные. В зависимости от степени детализации вайрфреймы могут включать элементы:

- навигация – хедер, футер, хлебные крошки, навигация по каталогу;
- заголовки и области текста – иногда их заменяют рыбным текстом или серыми прямоугольниками;
- кнопки – прямоугольники серого или белого цвета;
- карточки – блоки серого цвета, которые содержат в себе более мелкие элементы, например, прямоугольники текста и кнопок;
- иллюстрации и фотографии – вместо них ставят заглушки, например, перечёркнутые прямоугольники без заливки;
- иконки – в виде схемы или серых квадратов.

Разработка варфреймов направлена на визуализацию структуры и логики взаимодействия с веб-сервисом «МойРацион» до этапа детального дизайна. Основная цель – прототипирование ключевых пользовательских сценариев для обеспечения интуитивного и эффективного интерфейса, соответствующего задачам персонализации и автоматизации, заложенным в проекте. На рисунке 2 можно увидеть общую структуру варфреймов, где темно-серым цветом оттенка #646368 показаны расположения будущих изображений, картинок. Словами «Текст» обозначены будущие заголовки и описания сайта. Светло-серым отображены кнопки и поля форм на варфреймах в оттенке #9F9F9F. Варфреймы предполагают наличие в будущем дизайне модальных окон и слайдеров, которые также отражены в Приложении Б.

Разработанные экраны покрывают основной путь пользователя от первого знакомства до повседневного использования сервиса:

1) Стартовый экран: представляет ценностное предложение сервиса с акцентом на ключевые преимущества и содержит призыв к началу работы.

2) Экран входа и регистрации: обеспечивает минималистичный и быстрый процесс авторизации или создания аккаунта.

3) Многоэтапная анкета пользователя: реализует пошаговый сбор данных для глубокой персонализации.

4) Главный экран: выполняет роль центрального места, предоставляя сводку по дню, быстрый доступ к текущему плану питания и основным разделам сервиса, а также визуализацию прогресса.

5) Экран плана питания: детально отображает персонализированный рацион на выбранный день с возможностью гибкой замены блюд и просмотра рецептов.

6) Экран образовательного модуля: предоставляет интерфейс для просмотра статей.

7) Экран списка покупок: автоматически формирует и структурирует список необходимых продуктов на основе плана питания, позволяя пользователю управлять им, а также находить удобные для пользователя по расположению магазины на карте.

Разработанные варфреймы формируют целостную каркасную модель интерфейса, которая наглядно транслирует ключевые функциональные требования сервиса в логичные и удобные пользовательские потоки. Этот прототип служит основой для последующих этапов создания визуального дизайна и фронтенд-разработки, обеспечивая фокус на юзабилити и достижении целей проекта.

2.4 Mockup

Mockup (макет) – это высокодетализированное статическое изображение будущего интерфейса. На этом этапе к утвержденным вайрфреймам применена дизайн-система: добавлены цвета, типографика, иконография, реальный контент и текстуры. Цель данного этапа – визуализировать окончательный вид продукта, утвердить стилистическое единство всех экранов и проработать детали взаимодействия для усиления восприятия ключевых ценностей сервиса: персонализации, простоты и научного подхода [12].

Визуальный дизайн строится на принципах минимализма, ясности и эмпатии. Основной задачей является создание спокойной и сфокусированной среды, которая не перегружает пользователя, а помогает ему концентрироваться на достижении личных целей. Это достигается за счет:

– продуманной композиции и сетки, обеспечивающей четкую информационную иерархию и упорядоченность контента;

– интуитивной навигации, где ключевые действия выделены визуально, а второстепенные элементы не конкурируют за внимание.

Визуальное воплощение ключевых экранов напрямую следует из их функционального назначения и логики пользовательского сценария. Стартовый экран и авторизация выполнены максимально лаконично. Акцент сделан на фоновой иллюстрации, ассоциирующейся со свежестью и здоровьем, и четких формах. Это формирует положительное первое впечатление и устраняет барьеры на этапе входа. Стартовый экран содержит промо блок, описывающий основную суть веб-сервиса, блок преимуществ и блок отзывов. Кнопки призыва к авторизации расположены в достаточном количестве, чтобы пользователь свободно мог найти их. Рассмотреть Стартовый экран, экран входа и экран регистрации можно на рисунке 7.

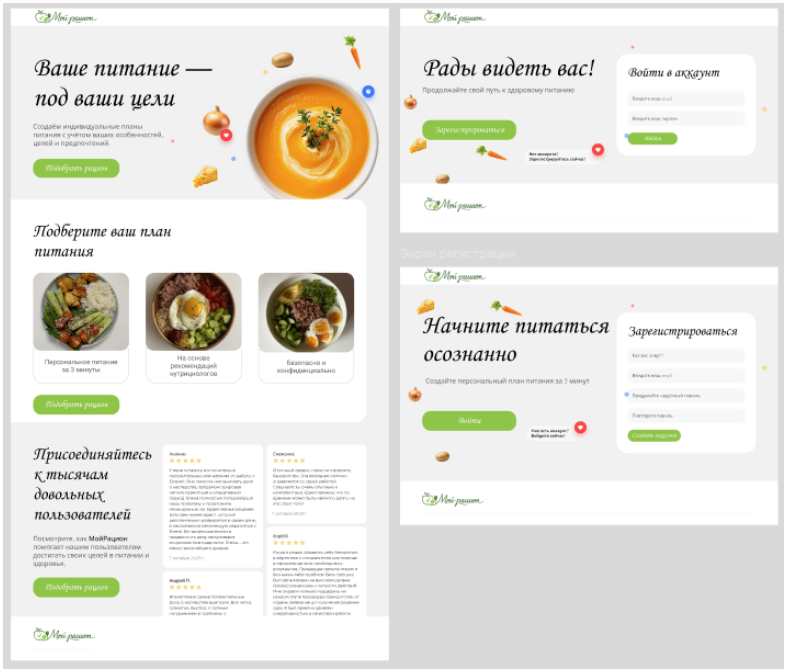

Рисунок 7 – Mockup экранов Старта, входа и регистрации

Многоэтапная анкета использует визуальные метафоры и интерактивные элементы для превращения процесса сбора данных из рутины в вовлекающее взаимодействие. Едва заметные, не отвлекающие пользователя от процесса цифры шагов постоянно показывает пользователю его продвижение, снижая ощущение сложности. Пастельные иллюстрации снижают тревожность пользователя и помогают легче осознать суть вопроса. Пример одного из семи шагов пользовательской анкеты расположен на рисунке 8.

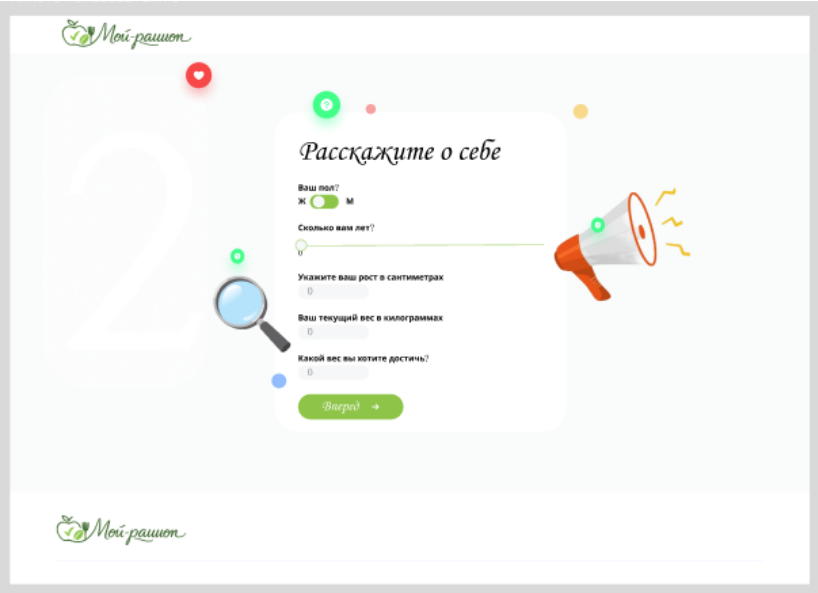

Рисунок 8 – Пример дизайна пользовательской анкеты

Главный экран спроектирован как сборник основных блоков. Информация о дневной норме КБЖУ, водном балансе и прогрессе представлена в виде наглядного графика, делая абстрактные данные осязаемыми. Главный слайдер содержит краткую выжимку из меню пользователя, при этом оставляя возможность перейти как к более подробному просмотру плана питания, так и к сформированному списку покупок. Для удобства пользователя настройки и слайдер со статьями, также располагаются прямо на главной странице, создавая у пользователя ощущение доступности ко всему.

Экран плана питания использует принцип модульности. Каждый прием пищи представлен как отдельная «умная» карточка с изображением блюда, что добавляет эстетики и аппетитности. Слайдер в первого блока кратко резюмирует недельное питание пользователя (рисунок 9).

Образовательная часть представлена слайдером на главном экране, а также возможностью перейти на экран самой статьи, прочитать и выбрать следующую (рисунок 6).

Список покупок представлен через лаконичный список с возможностью добавления, или удаления продуктов, а также картой с отмеченными магазинами на ней (рисунок 6).

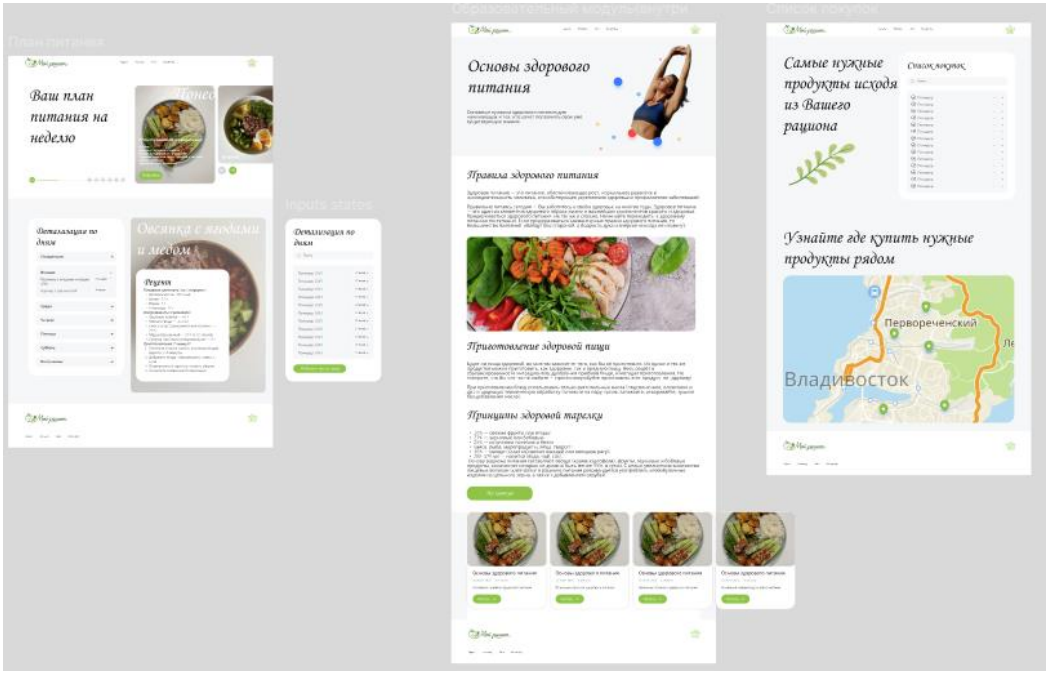

Рисунок 9 – Mockup экранов плана питания, образовательного модуля и списка покупок

Разработанные макеты представляют собой не просто набор оформленных экранов, а целостную визуальную экосистему, напрямую работающую на реализацию ценностного предложения «МойРацион». Ознакомится со всеми экранами макета можно в приложении В. Каждый дизайнерский выбор – от метафоры логотипа до расположения кнопки – подчинен общей цели: создать у пользователя ощущение контроля, ясности и уверенности на пути к своим целям в питании. Дизайн выступает мостом между сложной технологической начинкой и простым, человечным опытом взаимодействия, что является ключевым фактором для вовлечения и долгосрочной лояльности. Данные макеты утверждают финальный визуальный облик продукта и служат точкой отсчета для верстки и фронтенд-разработки.

2.5 Кликабельный прототип

Кликабельный прототип – это интерактивная модель, связывающая статические макеты в единый поток, который можно «пощупать». Главная задача прототипов – показать работу и будущего приложения или другой системы и на раннем этапе диагностировать возможные проблемы на пути бизнес – и пользовательских целей. Инструмент позволяет проверить юзабилити разрабатываемого продукта до начала написания кода, а значит сократить сроки и бюджет разработки. В прототипы гораздо проще вносить изменения, чем в уже готовые приложения. С помощью кликабельного прототипа можно наглядно и схематично увидеть сценарии и user flow приложения, способствующие достижению пользователем своих целей. Зачастую прототипы приводят к неожиданным открытиям и новым идеям, которые могут вывести проектируемый продукт на новый уровень, демонстрируют работу приложения в динамике, и мы можем оценить состояние потока (flow), то есть все ли идет гладко, быстро и понятно [13].

Разработка кликабельного прототипа является завершающим этапом проектирования пользовательского опыта и представляет собой интерактивную модель ключевых сценариев использования веб-сервиса «МойРацион». Целью создания прототипа комплексная валидация логики взаимодействия, навигационной структуры и юзабилити-гипотез до начала трудоемкой фазы разработки. Интерактивный прототип позволил перейти от статических макетов к симуляции реального использования, что критически важно для выявления и устранения потенциальных проблем на ранней стадии, ознакомится с ним можно отсканировав кьюаркод в Приложении Г.

Основной задачей прототипирования моделирование полного пользовательского пути нового пользователя: от первого знакомства со стартовым экраном до выполнения ежедневной рутины в личном кабинете. Ключевые сценарии, реализованные в прототипе, включали:

- Ознакомление с ценностным предложением и регистрацию.
- Прохождение многоэтапной анкеты для первоначальной персонализации.
- Взаимодействие с основными функциями личного кабинета: просмотр плана питания, изучение образовательного контента и работа со списком покупок.

Для сборки прототипа использовался специализированный инструмент для дизайна и прототипирования интерфейсов Figma. На рисунке 10 тонкими нитями показаны переходы по макету, по которму можно оценить количество переходов и возможностей пользователя.

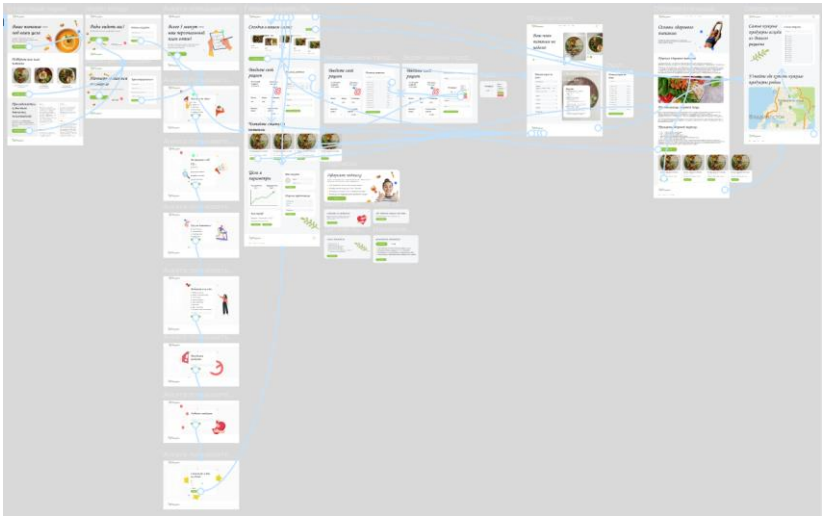

Рисунок 10 – Переходы кликабельного прототипа

Прототип интегрировал разработанные ранее статические макеты в единый интерактивный поток. Детально проработаны и реализованы все основные переходы:

- Линейные потоки, такие как пошаговое прохождение анкеты с навигацией «вперед/назад» и визуальным отражением прогресса.
- Непрерывные взаимодействия в рамках личного кабинета: переход между разделами через главное меню, открытие детализированных экранов.

- Микровзаимодействия, имитирующие реакцию интерфейса.

Навигационная модель строилась по принципу «центрального хаба» (личный кабинет), от которого пользователь может за несколько кликов добраться до любой ключевой функции. Это обеспечивало ощущение предсказуемости и контроля над интерфейсом.

Разработка кликабельного прототипа веб-сервиса «МойРацион» завершила этап проектирования пользовательского интерфейса и обеспечила переход к стадии технической реализации. Созданный прототип выполняет функцию детализированного технического задания, формализующего логику работы интерфейса, навигационные потоки и поведение элементов для команды разработки. Это минимизирует риски неверной интерпретации статических макетов и сокращает количество уточнений в процессе реализации. Кроме того, прототип позволил провести валидацию ключевых бизнес-сценариев, подтвердив корректность процессов онбординга, структуры личного кабинета и взаимосвязи его модулей с заявленным ценностным предложением.

Таким образом, интерактивная модель служит основным связующим артефактом между концептуальным проектированием и практической разработкой рабочего продукта.

3 Юзабилити-тестирование прототипа

Юзабилити тестирование – анализ того, как человек с помощью интерфейса решает задачу. Задачу определяет исследователь. Этот метод помогает находить закономерности в поведении пользователя и оценивать качество интерфейса – например, найти проблемы, с которыми пользователь сталкивается в рамках конкретного сценария [14]. Юзабилити-тестирование кликабельного прототипа веб-сервиса «МойРацион» проводилось с целью валидации разработанных пользовательских сценариев, оценки интуитивности интерфейса и выявления потенциальных проблем взаимодействия до начала этапа разработки. Тестирование организовано как серия структурированных сессий с представителями целевой аудитории.

Тестирование проводилось по методологии модерируемого тестирования «мыслей вслух» [15]. Привлечено 6 участников, соответствующих основным сегментам целевой аудитории: 2 представителя сегмента «занятые профессионалы», 2 – «начинающие энтузиасты ЗОЖ» и 2 – «целевые приверженцы ЗОЖ». Возраст участников – от 22 до 34 лет.

Каждая сессия длилась 20-30 минут и включала выполнение пяти ключевых задач, охватывающих основной пользовательский путь:

- регистрация в сервисе и первичное заполнение профиля;
- прохождение многоэтапной анкеты для настройки персонального плана питания;
- поиск и изучение плана питания на текущий день;
- использование функции «Список покупок» на основе предложенного плана;
- нахождение и просмотр материала в образовательном модуле.

Особое внимание уделялось фиксации времени выполнения задач, количеству ошибок, вербальным реакциям и невербальным сигналам.

Анализ данных тестирования позволил выявить как сильные стороны прототипа, так и критические точки, требующие доработки.

Положительные результаты:

- высокая оценка общего впечатления: 5 из 6 участников отметили современный и «чистый» визуальный стиль интерфейса;
- успешное выполнение основных задач: все участники самостоятельно выполнили задачи по просмотру статьи, добавление продукта; среднее время выполнения этих задач соответствовало ожидаемому.
- понятность базовой навигации: навигационное меню в личном кабинете признано интуитивным, переходы между разделами не вызывали затруднений.

Затруднения на этапе анкеты. Проблема: 4 из 6 пользователей «застревали» на 4-м шаге анкеты («Пищевые ограничения и аллергии»). Интерфейс с чекбоксами для выбора ограниче-

ний воспринят как избыточный и вызывающий тревогу («Я не болен, у меня просто непереносимость лактозы»). Отклонение: вместо последовательного выбора, пользователи начинали пропускать этот шаг или выбирали пункт «Нет ограничений», что могло привести к некорректной персонализации в дальнейшем. Обратная связь: участники предложили использовать более мягкие формулировки («Особенности питания») и добавить поле для свободного комментария. С исправленным вариантом с учетом обратной связи пользователей, можно ознакомится на рисунке 11.

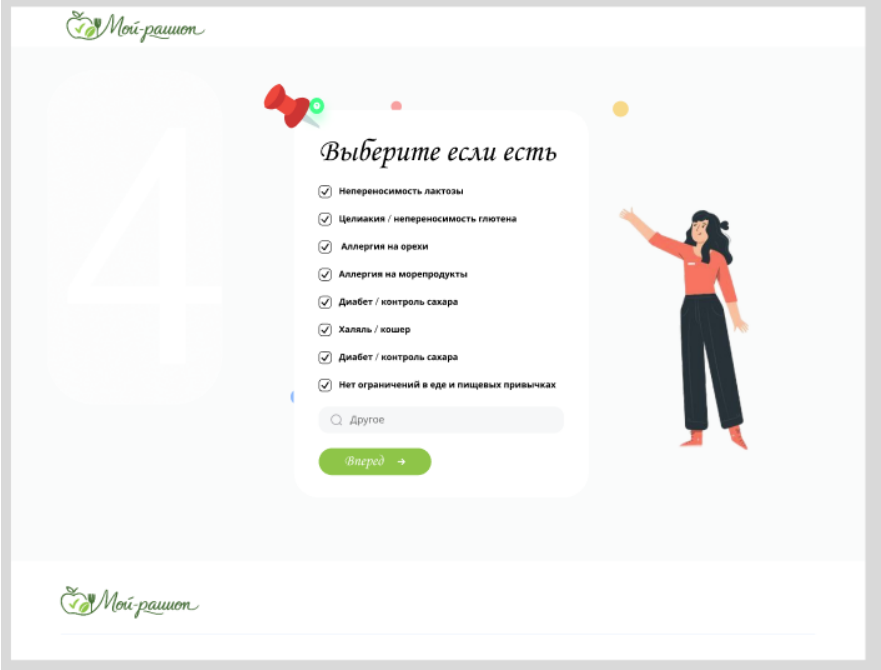

Рисунок 11 – Исправленный вариант Анкеты пользователя

Отсутствие возможности отметить купленным в разделе «Список покупок». Проблема: после формирования списка 4 пользователя не понимали, как отметить товары как купленные. Отсутствовала такая операция. Обратная связь: участники запросили чекбоксы для отметки покупки. С исправленным вариантом с учетом обратной связи пользователей, можно ознакомится на рисунке 12.

Проведенное юзабилити-тестирование доказало свою высокую эффективность как инструмент валидации проектных решений. Несмотря на положительную общую оценку эстетики и логики сервиса, тестирование выявило несколько критических проблем юзабилити, связанных с семантикой формулировок и отсутствием ключевых.

Анализ поведенческих паттернов пользователей подтвердил важность принципа «гибкого пути» – даже при четко спроектированном основном сценарии необходимо предусматривать альтернативные маршруты для исправления ошибок и изменения ранее введенных данных. Выявленные «точки застревания» не были случайными: они систематически возникали в местах, где дизайн-решение предполагало более высокий уровень цифровой грамотности или

мотивации, чем демонстрировала реальная аудитория. Это указывает на необходимость постоянной калибровки интерфейса под фактическое, а не предполагаемое поведение пользователей.

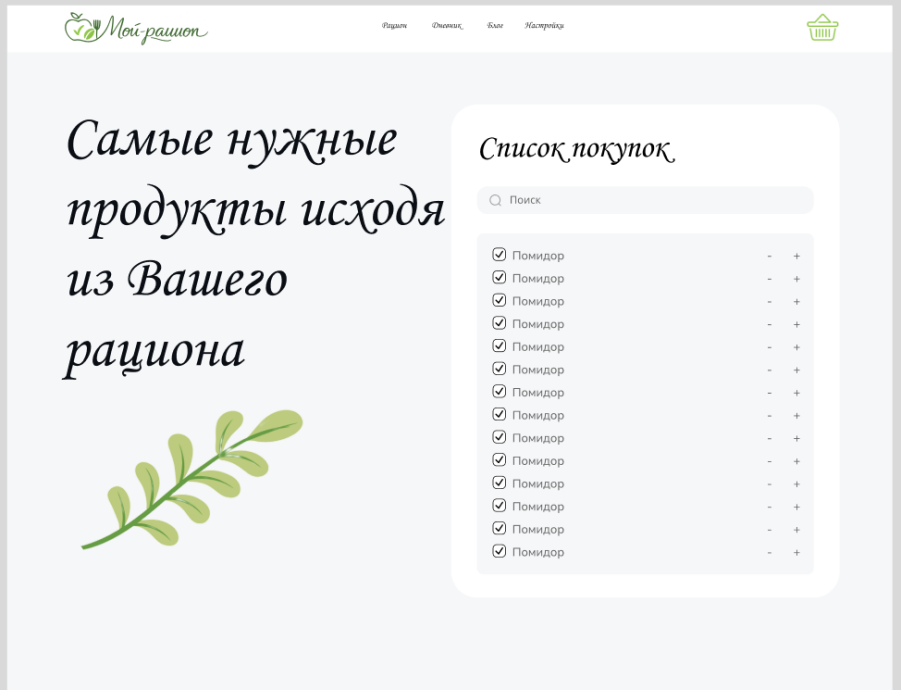

Рисунок 12 – Исправленный вариант Списка покупок

Все выявленные проблемы носили корректируемый характер и не требовали пересмотра базовой архитектуры интерфейса. Внесенные на основе фидбека изменения существенно повысили понятность и предсказуемость интерфейса, сократив потенциальное время на обучение и количество ошибок пользователей. Прототип, доработанный по итогам тестирования, можно считать валидированным и готовым к использованию в качестве точного и детализированного руководства для фронтенд-разработки.

Заключение

В ходе выполнения работы системно реализован полный цикл подготовки к разработке веб-сервиса «МойРацион», начиная от выбора технологического стека и заканчивая созданием и тестированием интерактивного прототипа. Результаты работы подтверждают реализуемость проекта и его соответствие современным требованиям к цифровым продуктам.

На первом этапе проведён сравнительный анализ доступных инструментов разработки, в результате чего обоснован выбор технологического стека, обеспечивающего оптимальное сочетание производительности, скорости разработки и масштабируемости. Выбранные платформы, фреймворки и средства управления версиями формируют надёжную основу для реализации функциональных требований сервиса.

Второй этап посвящён проектированию пользовательского опыта. На основе анализа целевой аудитории разработаны ключевые пользовательские сценарии, отражающие реальные потребности пользователей. Последовательно созданные варфреймы, визуальные макеты (Mockup) и кликабельный прототип позволили визуализировать логику взаимодействия с сервисом и проверить удобство навигации. Проведённое юзабилити-тестирование подтвердило интуитивность интерфейса и выявило точки для дальнейшей оптимизации.

Результаты работы свидетельствуют о том, что проект «МойРацион» обладает чёткой архитектурой, проработанным дизайном и подтверждённым удобством интерфейса. Все подготовительные этапы выполнены, что создаёт прочную основу для перехода к следующей фазе – непосредственной реализации функциональности веб-сервиса. Проект готов к запуску в разработку с минимальными рисками и чётким пониманием итогового продукта.

Список использованных источников

1 Cloud4y – Текст: электронный // Кто такой облачный провайдер и зачем он нужен? URL: https://www.cloud4y.ru/blog/what-is-a-cloud-provider/ (дата обращения: 11.12.2025)

2 Академия Selectel – Текст: электронный // Docker: что это такое и как работать с контейнерами URL: https://selectel.ru/blog/what-is-docker/ (дата обращения: 11.12.2025)

3 Dmosk – Текст: электронный // Что такое Podman URL: https://www.dmosk.ru/terminus.php?object=podman (дата обращения: 11.12.2025)

4 SkillFactory – Текст: электронный // Python URL: https://blog.skillfactory.ru/glossary/python/ (дата обращения: 11.12.2025)

5 Википедия – Текст: электронный // Система управления версиями URL: https://ru.wikipedia.org/wiki/ Система_управления_версиями (дата обращения: 11.12.2025)

6 Практикум – Текст: электронный // Что такое Git и для чего он нужен программистам – URL: https://practicum.yandex.ru/blog/chto-takoe-git-i-dlya-chego-nuzhen/ (дата обращения: 13.12.2025)

7 Skillbox – Текст: электронный // Что такое UX/UI-дизайн и как попасть в эту профессию. URL: https://skillbox.ru/media/design/ux_ui_dizayn_chto_eto_takoe/ (дата обращения: 11.12.2025)

8 Skillbox – Текст: электронный // Как выбрать и создать цветовую схему для сайтаURL: https://skillbox.ru/media/design/kak_vybrat_i_sozdat_tsvetovuyu_skhemu_dlya_sayta/ (дата обращения: 11.12.2025)

9 Semantica – Текст: электронный // Что такое логотип на сайте: размеры и размещение лого URL: https://semantica.in/blog/chto-takoe-logotip-na-sajte.html (дата обращения: 11.12.2025)

10 Практикум – Текст: электронный // Что такое User Flow и как его создать – URL: https://practicum.yandex.ru/blog/chto-takoe-user-flow-i-zachem-ego-razrabatyvat/ (дата обращения: 13.12.2025)

11 Практикум – Текст: электронный // Что такое вайрфреймы: собираем схему проекта– URL: https://practicum.yandex.ru/blog/chto-takoe-vajrfrejmy/ (дата обращения: 14.12.2025)

12 Практикум – Текст: электронный // Что начинающему дизайнеру нужно знать о мокапах – URL: https://practicum.yandex.ru/blog/chto-takoe-mokap/ (дата обращения: 15.12.2025)

13 Хабр – Текст: электронный // Чем так хороши кликабельные прототипы – URL: https://habr.com/ru/articles/692414/ (дата обращения: 16.12.2025)

14 Wannabe – Текст: электронный // Простые исследования: юзабилити-тестирование с прототипом – URL: https://library.wannabe.ru/article/prostye-issledovaniya-yuzabiliti-testirovanie-s-prototipom (дата обращения: 16.12.2025)

15 UxJournal – Текст: электронный // Что такое юзабилити-тестирование и как провести его правильно – URL: https://ux-journal.ru/chto-takoe-yuzabiliti-testirovanie-i-kak-provesti-ego-pravilno.html (дата обращения: 16.12.2025)

Приложение А

Пользовательские сценарии

Рисунок А.1 – кьюаркод на User Flow Map

Приложение Б

Wireframes

Рисунок Б.1 – Wireframes

Приложение В

Mockup

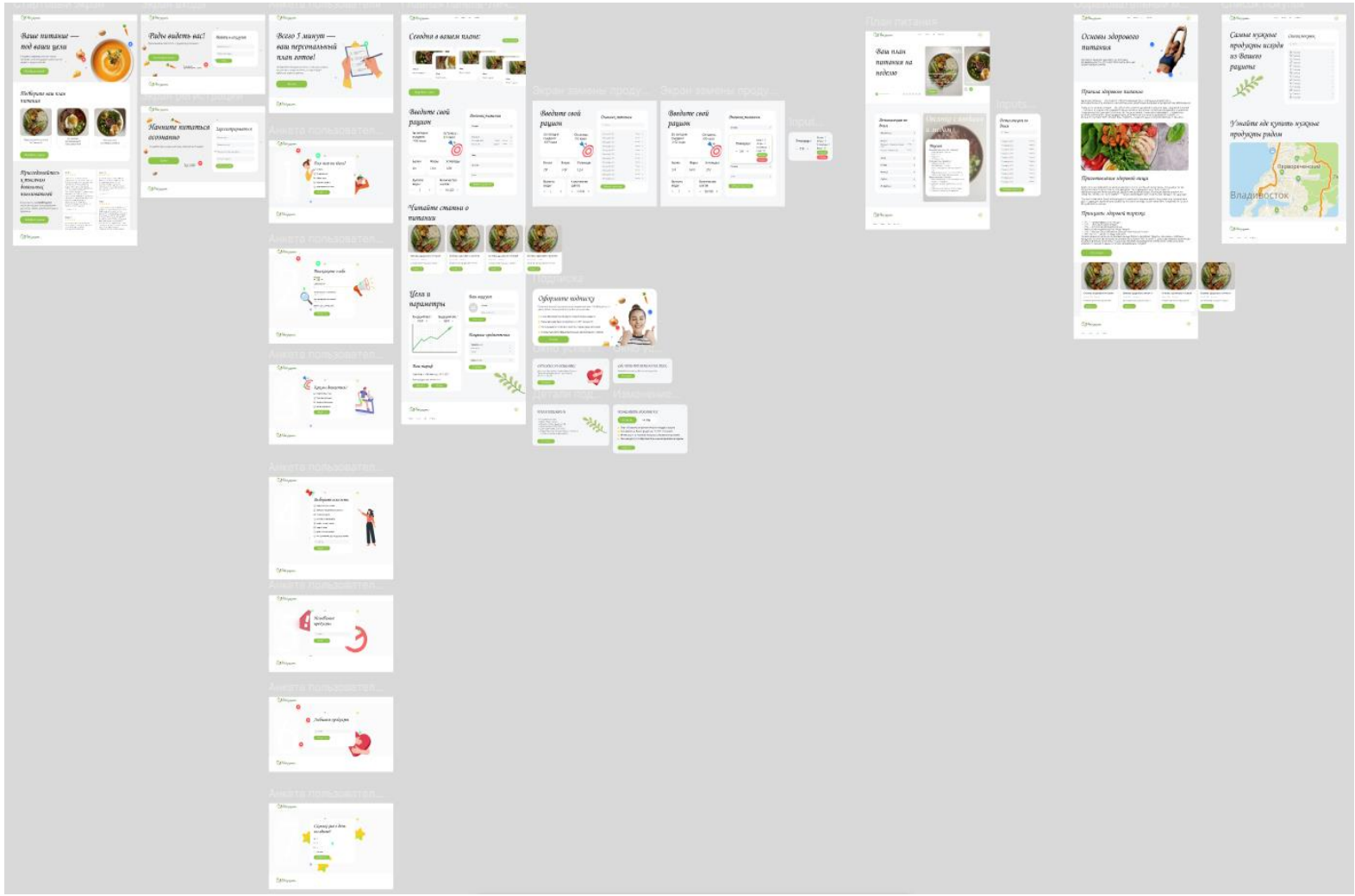

Рисунок В.1 – Mockup

Приложение Г

Кликабельный прототип

Рисунок Г.1 – Кьюаркод на кликабельный прототип