|  |  |  |  |
| --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН  СЕВЕРО-КАЗАХСТАНСКИЙ УНИВЕРСИТЕТ ИМ. М. КОЗЫБАЕВА | | | |
|  | | | |
|  | | |  |
| СЕМЕСТРОВЫЙ ПРОЕКТ  На тему: «Nocode приложение «partyparty!» | | | |
| Выполнили студенты  группы АПО-20-2 |  | Суиндиков М.  Степаноав Р.  Чучман А.  Ставская Е. | |
| Доктор PhD,  доцент кафедры ИКТ |  | Астапенко Н.В. | |
|  |  |  | |

Петропавловск, 2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc133925196)

[1 Аналитическая часть 4](#_Toc133925197)

[1.1 Актуальность проекта 4](#_Toc133925198)

[1.2 Перспективы и финансовая выгода 4](#_Toc133925199)

[1.3 Структурно-логическое моделирование 5](#_Toc133925200)

[1.4 Методология разработки Scrum 7](#_Toc133925201)

[1.6 Плюсы и минусы разработки на Nocode 11](#_Toc133925202)

[1.6 Технология Bubble.IO 14](#_Toc133925203)

[2 Проектная часть 16](#_Toc133925204)

[2.1 Разработка UserStory 16](#_Toc133925205)

[2.2 Проектирование пользовательских интерфейсов 18](#_Toc133925206)

[ЗАКЛЮЧЕНИЕ 24](#_Toc133925207)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 25](#_Toc133925208)

ВВЕДЕНИЕ

В поиске тиммейтов для игр на различных платформах важно иметь сильную команду, которая будет взаимодействовать вместе и достигать общих целей. Найти подходящих людей для игры на различных платформах может быть сложной задачей, особенно если у вас ограниченное количество времени. В связи с этим, в рамках данного проекта было разработано no-code приложение, которое позволяет найти искомых союзников для дальнейшей взаимовыручки.

Целью проекта является разработка no-code приложения «partyparty!», которое позволяет быстро создать профиль и указать игровую платформу, игровые навыки, локацию и другие критерии, которые необходимы для поиска подходящих тиммейтов. Это делает процесс поиска команды для игры гораздо проще и эффективнее.

Функционал приложения включает в себя регистрацию и авторизацию, а также возможность изменения профиля и импорта данных из социальных сетей. Приложение также предоставляет возможность лайкать и дизлайкать анкеты других пользователей, чтобы помочь сужать круг поиска и находить подходящих тиммейтов быстрее. Кроме того, есть возможность заходить в профили других пользователей и использовать функционал чата, чтобы связаться с ними и обсудить возможное сотрудничество.

Объектом исследования данного проекта является no-code приложение для эффективного поиска тиммейтов. Предметом исследования являются функциональные возможности приложения, ее эффективность и удобство использования для конечного пользователя.

В процессе работы были использованы следующие методы исследования: анализ литературы по теме, изучение аналогов, разработка no-code приложения.

Практическая значимость данного проекта заключается в создании удобного и эффективного приложения для поиска тиммейтов, которое поможет пользователям быстро и легко найти подходящих тиммейтов на основе их интересов, навыков.

# 1 Аналитическая часть

1.1 Актуальность проекта

Актуальность данного проекта заключается в том, что он решает проблему поиска партнеров для совместной игры на различных игровых платформах. Социальный аспект игр на сегодняшний день очень важен, так как игры стали не только средством развлечения, но и способом социализации. Однако, не всегда легко найти подходящих партнеров для совместной игры.

Наше приложение предоставляет возможность геймерам найти и подобрать партнеров для совместной игры на основе их игровых предпочтений, навыков и локации. Оно помогает игрокам находить единомышленников, которые разделяют их интересы и увлечения, и создавать крепкие команды для совместного прохождения игровых заданий.

Кроме того, наше приложение позволяет расширить социальную сеть и найти новых друзей, что важно для геймеров, которые могут чувствовать себя одинокими в своих интересах.

Важно отметить, что приложение создается с использованием no-code методологии, что означает быстрое и эффективное развитие проекта без необходимости тратить время и ресурсы на написание кода. Это позволяет нам быстро адаптироваться к потребностям пользователей и обеспечивать их лучшим игровым опытом.

Таким образом, наше приложение является актуальным решением для геймеров, которые ищут партнеров для совместной игры и желают расширить свою социальную сеть.

1.2 Перспективы и финансовая выгода

Перспективы данного проекта связаны с тем, что он обладает большим потенциалом для расширения пользовательской базы. Игры стали неотъемлемой частью культуры молодежи, и с каждым годом количество игроков только увеличивается. Также игроки все больше и больше нуждаются в совместных игровых опытах и компаньонах для игры. В связи с этим, спрос на подобные приложения будет только расти, что делает проект перспективным.

Одним из способов получения финансовой выгоды из данного проекта может стать реклама игровых продуктов и услуг внутри приложения. Компании, которые занимаются разработкой и продвижением игр, могут использовать нашу платформу для привлечения новых игроков на свои игровые проекты. Также мы можем предлагать платные услуги, такие как расширенный функционал для профиля, возможность более широкой аудитории видеть ваш профиль и т.д.

В дополнение к этому, мы можем предоставлять статистические данные и аналитику по игровым тенденциям и предпочтениям пользователей нашей платформы, что может заинтересовать разработчиков игр, которые хотят получить дополнительную информацию о потенциальных пользователях своих продуктов.

В целом, проект имеет перспективы для роста и развития, и при правильном подходе может приносить значительную финансовую выгоду. Однако необходимо учитывать, что конкуренция в данной сфере достаточно высока, поэтому для успеха проекта необходимо уделить внимание качеству и функциональности приложения, а также правильному маркетинговому подходу.

1.3 Структурно-логическое моделирование

IDEF0 (Integration Definition for Function Modeling) – это методология структурно-логического моделирования, которая используется для анализа, проектирования и документирования функций бизнес-процессов. Она позволяет описать взаимодействие компонентов системы и определить их функциональные зависимости.

В первую очередь была проведена декомпозиция проекта на блоки, каждый из которых отвечает за выполнение определённой задачи. Далее были определены связи между этими блоками, которые выполняют процессы в приложении.

При моделировании схемы были выделены следующие функциональные блоки:

* проектирование;
* разработка;
* тестирование и отладка;
* развёртывание и сопровождение.

Каждый из этих функциональных блоков является важным этапом в разработке приложения и требует специфических навыков и знаний от разработчиков. Важно понимать, что успешная разработка приложения не зависит только от одного блока, а требует взаимодействия всех блоков вместе.

Диаграмма в нотации IDEF0 представлена в соответствии с рисунком 1.

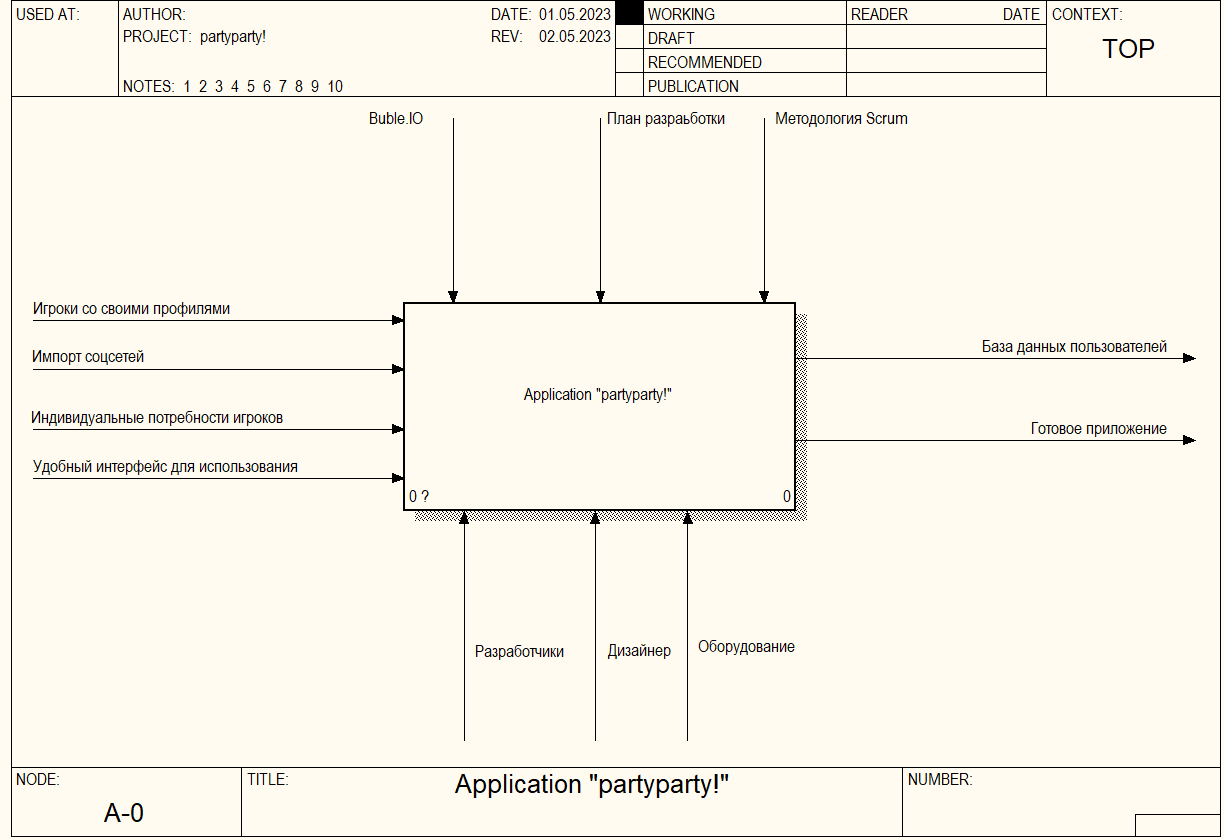


Рисунок 1 Диаграмма IDEF0

Декомпозиция IDEF0 начинается с выделения основной функциональной блок-схемы, которая представляет собой верхнеуровневое описание основных функций системы. Затем этот блок разбивается на более мелкие блоки, называемые функциональными блоками первого уровня. Каждый функциональный блок первого уровня может быть детально описан в соответствующей функциональной блок-схеме. Этот процесс продолжается до тех пор, пока не будут достигнуты достаточно мелкие функциональные блоки, которые могут быть подробно описаны в соответствующей функциональной блок-схеме. Декомпозиция диаграммы IDEF0 представлена в соответствии с рисунком 2.

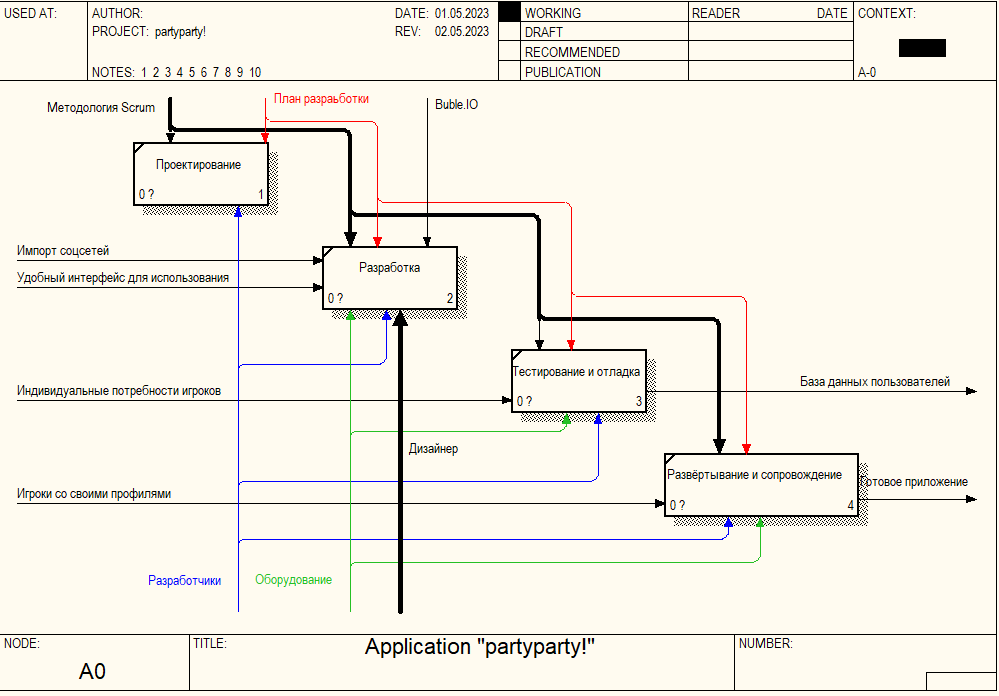


Рисунок 2 Декомпозиция IDEF0

IDEF0 – это мощный инструмент, позволяющий разработчикам и экспертам описывать, анализировать и улучшать бизнес-процессы в организации. Он помогает улучшить понимание того, как работает организация, и выявить возможные проблемы, что в свою очередь может привести к более эффективной работе компании.

1.4 Методология разработки Scrum

Scrum – это гибкая методология разработки программного обеспечения, которая обеспечивает эффективное управление проектами и повышение качества продукта. Она основывается на итеративной и инкрементальной модели разработки, которая позволяет быстро и гибко реагировать на изменения требований и обеспечить доставку ценных продуктов в короткие сроки.

Основные принципы Scrum:

* Команда самоорганизовывается и сама принимает решения, необходимые для выполнения работы.
* Весь процесс разработки разбивается на короткие итерации (спринты), которые продолжительностью обычно 2-4 недели.
* Каждый спринт начинается с планирования, в котором определяются цели и задачи на следующий период.
* В течение спринта команда ежедневно проводит краткие совещания, на которых обсуждается текущий прогресс работы и проблемы, которые необходимо решить.
* По окончании спринта проводится демонстрация результатов и ретроспектива, на которых команда обсуждает результаты работы и определяет области для улучшения.

Плюсы Scrum:

* Гибкость: Scrum предоставляет возможность гибко управлять проектом, позволяя команде быстро реагировать на изменения в требованиях и условиях проекта.
* Прозрачность: Scrum обеспечивает прозрачность в процессе разработки, позволяя всей команде видеть текущее состояние проекта и принимать правильные решения на основе этой информации.
* Участие заказчика: Scrum предполагает активное участие заказчика в процессе разработки, что позволяет более точно понимать его требования и удовлетворять его потребности.
* Снижение рисков: Scrum помогает снизить риски и повысить качество продукта, разбивая проект на меньшие итерации и обеспечивая частые проверки и обратную связь.
* Мотивация команды: Scrum способствует мотивации команды, обеспечивая ей возможность принимать участие в принятии решений и управлении проектом.

Недостатки Scrum:

* Недостаточная документация: Scrum подразумевает уменьшение объема документации, что может привести к недостаточному пониманию требований и проблем при передаче проекта другой команде.
* Сложность обучения: Scrum может быть сложной для понимания и применения, особенно для новых членов команды, что требует дополнительного времени и усилий на обучение.
* Не подходит для всех проектов: Scrum может не быть подходящим для всех проектов, особенно для проектов с жесткими требованиями и графиками.
* Неэффективное управление: Неправильное управление Scrum может привести к проблемам в коммуникации и координации между членами команды.
* Неподходящий размер команды: Неправильный размер команды может привести к неэффективной работе, особенно в больших проектах, требующих сильной координации и коммуникации.

Одним из ключевых преимуществ Scrum является гибкость и возможность быстро реагировать на изменения требований и пожелания заказчика. Кроме того, методология Scrum способствует повышению эффективности работы команды, улучшению качества продукта и удовлетворению потребностей пользователей.

В то же время, Scrum может иметь некоторые недостатки, такие как сложность внедрения и поддержания процесса, особенно в больших командах, а также возможность неправильного определения приоритетов и планирования работы. Однако, эти недостатки могут быть преодолены с помощью правильной организации и использования методологии Scrum.

Существует множество методологий управления проектами, и каждая из них имеет свои преимущества и недостатки. Рассмотрим несколько популярных методологий и сравним их с Scrum.

* Водопадная методология: в этой методологии каждый этап проекта выполняется последовательно, начиная с планирования и заканчивая тестированием. Однако, этот подход не предусматривает гибкости и возможности внесения изменений в процессе выполнения проекта. Scrum же, наоборот, предлагает гибкое и итеративное планирование и управление проектом, что позволяет более эффективно реагировать на изменения.
* Каскадная методология: этот метод управления проектами похож на водопадную, но включает в себя более частые проверки процесса выполнения проекта и более гибкие механизмы управления изменениями. Однако, он все еще не обладает такой же степенью гибкости, как Scrum, и не предлагает столь же четкой структуры команды и обязанностей.
* Kanban: это методология, основанная на визуальном отслеживании задач, которые находятся в процессе выполнения, и управлении потоком работы. Однако, Kanban не обладает такими же четкими ролями и ответственностями, как Scrum, и может быть менее эффективным в управлении проектом, который требует высокой степени координации и синхронизации.

В пользу выбора Scrum могут говорить следующие аргументы:

* Гибкость: Scrum предлагает гибкий и адаптивный подход к разработке, что означает, что команда может быстро реагировать на изменения требований и потребностей клиента, что не всегда возможно с более традиционными методологиями разработки.
* Большая прозрачность: Scrum уделяет особое внимание коммуникации и прозрачности в рамках команды и с заказчиком, что делает процесс разработки более предсказуемым и понятным для всех сторон.
* Увеличение качества: Scrum подчеркивает значимость тестирования и обеспечения качества в процессе разработки, что в конечном итоге может привести к высококачественному продукту.
* Ускорение процесса: Scrum предлагает регулярные встречи и короткие сроки разработки, что способствует более быстрой доставке готового продукта на рынок
* Простота: Scrum довольно прост в понимании и реализации, что позволяет командам быстро начинать работу в этой методологии.

Хотя Scrum не является универсальным решением для всех проектов, он может быть очень эффективным во многих ситуациях, особенно в быстро меняющихся и динамичных окружениях разработки.

1.5 Технология Coda.io

Coda.io – это онлайн-сервис, который предоставляет возможность объединения функциональности электронных таблиц, баз данных, документов и других инструментов в едином рабочем пространстве. Пользователи могут создавать документы, проекты и таблицы с уникальными функциями, интегрировать их с другими приложениями и сервисами, а также работать над ними совместно с командой в режиме реального времени. Coda.io отличается гибкостью, универсальностью и удобством использования. Он позволяет создавать документы любой сложности, использовать формулы для создания любых необходимых функций, расширять функциональность с помощью сторонних интеграций и готовых шаблонов. Кроме того, Coda.io предоставляет возможность работать с документами в режиме реального времени с другими участниками команды, видеть изменения, вносимые другими участниками, и обсуждать их встроенными инструментами комментирования и обсуждения. Сервис также предоставляет множество интеграций со сторонними приложениями и сервисами, что позволяет автоматизировать множество задач и создавать более эффективные рабочие процессы. Кроме того, Coda.io сохраняет историю изменений и обеспечивает высокий уровень безопасности для хранения и обработки данных.

Coda.io – это инструмент для управления проектами и коммуникации внутри команды, который предоставляет широкие возможности для организации рабочего процесса. Рассмотрим некоторые плюсы и минусы Coda.io.

Плюсы Coda.io:

* Универсальность: Coda.io предоставляет широкие возможности для организации рабочего процесса и коммуникации внутри команды. Вы можете создавать таблицы задач, документы, таблицы коммуникации и использовать готовые шаблоны для создания проектов.
* Интеграция с другими приложениями: Coda.io интегрируется с другими приложениями, такими как Slack, Gmail, Google Drive и другими. Это упрощает работу и снижает затраты времени на переключение между различными приложениями.
* Гибкость: Coda.io позволяет настроить таблицы и документы под нужды вашей команды, в том числе добавлять новые поля и настраивать интерфейс для удобства работы.
* Доступность: Coda.io доступен как веб-приложение и имеет мобильные приложения для iOS и Android.

Минусы Coda.io:

* Сложность: Coda.io может быть сложным в использовании для новых пользователей, которые не имеют опыта работы с подобными инструментами.
* Стоимость: Coda.io предоставляет бесплатную версию, но для получения полного набора функций необходимо подписаться на платную версию.
* Не подходит для всех проектов: Coda.io может не подойти для больших и сложных проектов, где требуется более сложная система управления проектами.

Почему удобно использовать Coda.io:

* Широкие возможности: Coda.io предоставляет широкие возможности для организации рабочего процесса и коммуникации внутри команды.
* Интеграция с другими приложениями: Coda.io интегрируется с другими приложениями, что упрощает работу и снижает затраты времени на переключение между различными приложениями
* Гибкость: Coda.io позволяет настроить таблицы и документы под нужды вашей команды.
* Доступность: Coda.io доступен как веб-приложение и имеет мобильные приложения для iOS и Android, что позволяет работать в любое время и из любой точки мира.
* Организация общения: Coda.io позволяет создавать таблицы для коммуникации, где члены команды могут обмениваться идеями, задавать вопросы и обсуждать проблемы, связанные с проектом. Это позволяет сократить время на организацию встреч и упростить процесс общения внутри команды.
* Автоматизация: Coda.io позволяет автоматизировать рутинные задачи, например, отправку уведомлений о задачах, которые требуют выполнения.

В целом, Coda.io – это мощный инструмент для создания и управления документами и таблицами, который позволяет пользователям создавать настраиваемые рабочие процессы и работать с командой в режиме реального времени.

Наша команда использует Coda.io для управления задачами, сроками и коммуникацией внутри нашего проекта. Мы выбрали Coda.io, потому что это мощный инструмент, который упрощает работу над проектом и обеспечивает широкие возможности для управления проектами, коммуникации внутри команды и интеграции с другими приложениями. Также мы можем назначать задачи для каждого члена команды и отслеживать прогресс выполнения каждой задачи.

1.6 Плюсы и минусы разработки на Nocode

Разработка на nocode является достаточно новым и быстроразвивающимся направлением, которое позволяет быстро создавать прототипы и минимальные версии приложений без необходимости в глубоких знаниях программирования.

Один из главных плюсов разработки на nocode – это ускорение процесса разработки. В сравнении с разработкой на коде, разработка на nocode может занять намного меньше времени и затрат, что в свою очередь может привести к более быстрому выходу на рынок и ускорению обратной связи от пользователей.

Другой плюс разработки на nocode – это более простое и понятное управление приложением, что позволяет бизнес-пользователям быстрее внедрять идеи и реализовывать новые проекты.

Еще ряд плюсов nocode:

* Быстрый запуск проекта: благодаря использованию готовых элементов и компонентов, разработка на nocode может занять меньше времени и усилий, что позволяет быстрее запустить проект на рынок и начать получать обратную связь от пользователей.
* Легкость в использовании: разработка на nocode не требует глубоких знаний программирования, что делает этот метод доступным для широкого круга пользователей, включая менеджеров, маркетологов, дизайнеров и т.д.
* Экономия затрат: разработка на nocode может значительно сократить затраты на создание приложения, что особенно актуально для небольших компаний и стартапов.
* Гибкость: хотя некоторые платформы nocode могут быть ограничены в функциональности, многие из них позволяют настраивать приложения под конкретные потребности пользователей и команды.

Интеграция с другими сервисами: многие платформы nocode предоставляют интеграцию с другими сервисами и инструментами, что упрощает и ускоряет процесс разработки.

Однако, разработка на nocode имеет и свои минусы:

* Ограниченность функциональности: приложения, созданные на nocode, могут быть ограничены в своей функциональности, что может ограничить возможности для более сложных проектов.
* Низкая гибкость: при работе с nocode может быть ограничена гибкость в реализации определенных функций и возможности быстрой оптимизации приложения.
* Зависимость от платформы: приложения, созданные на nocode, зависят от платформы, которая используется для их создания. Если платформа перестанет поддерживаться, приложение может перестать работать.
* Ограниченный контроль над кодом: в отличие от создания приложения на коде, использование nocode не дает полного контроля над кодом и может ограничить возможности интеграции с другими системами.
* Невозможность настройки под сложные бизнес-процессы: приложения, созданные на nocode, могут быть недостаточно гибкими для настройки под сложные бизнес-процессы, которые могут потребовать индивидуальных подходов.
* Невозможность разработки кастомных решений: в некоторых случаях может потребоваться создание кастомных решений, которые невозможно реализовать на nocode.
* Ограниченный доступ к данным: некоторые платформы nocode могут иметь ограниченный доступ к данным, что может быть недостаточным для реализации некоторых функций.

Также важно отметить, что использование nocode не подходит для всех типов проектов, и в некоторых случаях может потребоваться более продвинутый подход.

Однако, некоторые известные компании, такие как Zapier, Webflow, Airtable и Bubble, успешно используют nocode в своих проектах.

Кроме того, nocode может быть полезен для стартапов и малых бизнесов, которые не имеют достаточных финансовых ресурсов для найма команды разработчиков и/или необходимого оборудования.

В целом, nocode – это достаточно новое направление в разработке приложений, которое имеет свои плюсы и минусы, и может быть эффективным для некоторых проектов и бизнес-моделей.

Использование nocode технологии для разработки приложений может представлять собой новый и уникальный опыт для команды разработчиков.

Во-первых, разработка на nocode платформах может открыть новые возможности для команды в создании приложений, особенно для тех, кто не имеет достаточного опыта в программировании. Это может привести к тому, что команда может создавать приложения быстрее и более эффективно, чем при использовании традиционных методов разработки.

Во-вторых, использование nocode технологии может позволить команде разработчиков углубиться в аспекты проектирования и пользовательского интерфейса, что может сделать приложение более интуитивно понятным и привлекательным для пользователей.

Третий важный аспект – это возможность экспериментировать с различными функциональностями и концепциями приложения без необходимости писать код. Это может ускорить процесс прототипирования и позволить команде быстро выявлять ошибки и улучшать приложение.

Однако, новый опыт в использовании nocode технологии также может представлять собой вызов для команды. Как и при любом новом опыте, возможны трудности и препятствия, такие как необходимость изучения новых функций и возможностей платформы, а также необходимость адаптации к новым рабочим процессам и методам разработки.

Есть и другие причины использования nocode технологии:

* Быстрые сроки разработки: если проект должен быть реализован быстро, то использование nocode может значительно сократить сроки разработки и ускорить запуск приложения на рынок. Это особенно важно для стартапов и компаний, которые стремятся занять лидирующие позиции на рынке.
* Простота использования: если команда не имеет достаточного опыта в программировании, то использование nocode может быть более простым решением для создания приложения. Готовые инструменты и платформы могут предоставлять удобный интерфейс для создания и настройки приложения, что позволяет бизнес-пользователям быстро реализовывать свои идеи без необходимости обращаться к разработчикам.
* Гибкость: Некоторые nocode платформы предоставляют большую гибкость при создании приложения. Например, платформа Bubble позволяет создавать приложения, которые могут быть интегрированы с другими системами и иметь сложную функциональность, такую как авторизация, платежи и многое другое.

В целом, использование nocode технологии для разработки приложений может представлять собой новый и уникальный опыт для команды разработчиков, который может привести к более эффективному и качественному процессу разработки приложений.

1.6 Технология Bubble.IO

Bubble.io – это платформа для создания веб-приложений без необходимости писать код. Вот некоторые преимущества данного сервиса:

* Низкий порог входа: Bubble.io имеет интуитивно понятный интерфейс и не требует знания программирования. Это делает его доступным для широкого круга пользователей, в том числе для тех, кто только начинает свой путь в веб-разработке.
* Быстрота разработки: благодаря множеству готовых элементов и возможности перетаскивания компонентов, разработка на Bubble.io происходит гораздо быстрее, чем при написании кода с нуля.
* Интеграции: Bubble.io позволяет легко интегрировать сторонние сервисы и API, что упрощает процесс разработки и расширяет функциональность создаваемого приложения.
* Низкая стоимость: использование Bubble.io позволяет сэкономить на затратах на разработку, так как нет необходимости нанимать команду программистов.

Однако, существуют и некоторые недостатки:

* Ограничения функциональности: приложения, созданные на Bubble.io, могут ограничиваться возможностями, предоставляемыми самой платформой, поэтому некоторые функции могут быть трудно реализовать.
* Производительность: в сравнении с приложениями, написанными на «чистом» коде, производительность приложений, созданных на Bubble.io, может быть ниже.
* Ограниченный контроль: при использовании Bubble.io пользователь не имеет полного контроля над создаваемым приложением и может столкнуться с ограничениями настройки.

Для разработки приложения был выбран Bubble по нескольким причинам:

* Простота использования: Bubble предоставляет пользовательский интерфейс, который позволяет создавать приложения без необходимости знать программирование. Это делает платформу доступной для широкой аудитории и позволяет быстро создавать прототипы приложений.
* Гибкость: Bubble позволяет настроить приложение под нужды команды. Это включает в себя создание пользовательских интерфейсов, настройку баз данных и логики приложения.
* Интеграция: Bubble имеет множество интеграций с другими приложениями и сервисами, что позволяет команде использовать различные инструменты для создания приложения.
* Экономическая выгода: Bubble предоставляет бесплатную версию, которая позволяет создать базовое приложение. Это может быть особенно полезно для небольших команд с ограниченным бюджетом.

Bubble.io используют как небольшие стартапы и некоммерческие проекты, так и крупные компании, которые быстро хотят вывести на рынок минимальный жизнеспособный продукт.

# 2 Проектная часть

# 2.1 Разработка UserStory

User Story 1: Регистрация

Как пользователь, я хочу зарегистрироваться в приложении, чтобы мне стало доступно все функциональные возможности, и я мог начать поиск товарищей по командным онлайн и кооперативным играм.

Действия пользователя:

* открыть приложение;
* нажать на кнопку «Регистрация»;
* заполнить форму регистрации, указав свои данные, включая логин, электронную почту и пароль;
* нажать на кнопку «Зарегистрироваться»;
* получить подтверждение о регистрации и перейти на страницу авторизации.

Ожидаемый результат:

Пользователь успешно зарегистрирован в приложении и может авторизоваться для использования всех функций.

User Story 2: Авторизация

Как зарегистрированный пользователь, я хочу авторизоваться в приложении, чтобы мне стало доступно все функциональные возможности, и я мог начать поиск товарищей по командным онлайн и кооперативным играм.

Действия пользователя:

* открыть приложение;
* ввести свой логин и пароль;
* нажать на кнопку «Войти».

Ожидаемый результат:

Пользователь успешно авторизован в приложении и может начать использовать все функциональные возможности.

User Story 3: Возможность материально поддержать проект

Как пользователь, я хочу иметь возможность материально поддержать проект, чтобы я мог получать дополнительные привилегии или функциональные возможности.

Действия пользователя:

* открыть приложение;
* перейти на страницу поддержки проекта;
* выбрать подходящий способ поддержки проекта;
* следовать инструкциям на экране для завершения процесса поддержки.

Ожидаемый результат:

Пользователь успешно поддержал проект и может использовать дополнительные привилегии и функциональные возможности.

User Story 4: Просмотр и оценивание анкет пользователей

Как пользователь, я хочу иметь возможность просмотреть анкеты других пользователей и оценить их навыки и опыт, чтобы я мог найти подходящего партнера для командной игры.

Действия пользователя:

* открыть приложение;
* перейти на страницу «Анкеты пользователей»;
* выбрать анкету, которую пользователь хочет посмотреть;
* просмотреть информацию об этом пользователе, включая его никнейм, игровые навыки, опыт, предпочитаемые игры, часовой пояс и другую информацию;
* оценить навыки пользователя на основе его профиля и предыдущих игр.

Ожидаемый результат:

Пользователь может просмотреть анкету другого пользователя и оценить его навыки и опыт, чтобы определить, подходит ли он для командной игры.

User Story 5: Поиск по запросу среди анкет пользователей

Как пользователь, я хочу иметь возможность найти других пользователей по запросу, чтобы я мог найти подходящего партнера для командной игры.

Действия пользователя:

* открыть приложение;
* перейти на страницу поиска пользователей;
* ввести запрос для поиска, включая название игры, никнейм игрока или часовой пояс;
* нажать на кнопку «Найти»;
* просмотреть результаты поиска.

Ожидаемый результат:

Пользователь может найти других пользователей по запросу и связаться с ними для игры.

User Story 6: Пользовательский чат

Как пользователь, я хочу иметь возможность общаться с другими пользователями через чат, чтобы я мог быстро договориться о командной игре.

Действия пользователя:

* открыть приложение;
* выбрать пользователя, с которым пользователь хочет общаться;
* написать сообщение в чат;
* ожидать ответа.

Ожидаемый результат:

Пользователь может общаться с другими пользователями через чат и быстро договориться о командной игре.

User Story 7: Редактирование пользовательской анкеты

Как пользователь, я хочу иметь возможность редактировать свою анкету, чтобы я мог обновлять свои навыки и опыт в играх.

Действия пользователя:

* открыть приложение;
* перейти на страницу своей анкеты;
* нажать на кнопку «Редактировать»;
* изменить информацию в своей анкете;
* нажать на кнопку «Сохранить изменения».

Ожидаемый результат:

Пользователь успешно отредактировал свою анкету и обновил свои навыки и опыт в играх.

# 2.2 Проектирование пользовательских интерфейсов

Проект имеет такие функции, как: «Регистрация и авторизация», «Чат», «Поиск анкет», «Оценка и просмотр анкет» и «Редактирование анкет».

При входе в приложение пользователя встречает окно загрузки. Окно загрузки представлена в соответствии с рисунком 2.1.

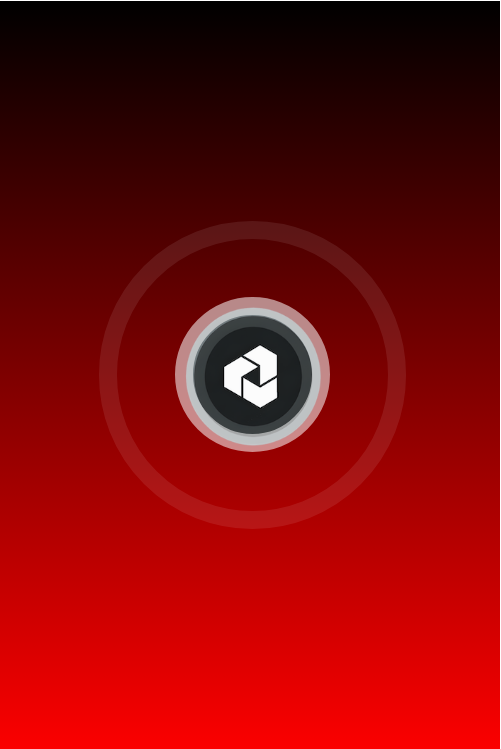


Рисунок 2.1 Окно загрузки

Далее следует окно Аутентификации пользователя с выбором авторизации если пользователь уже есть в системе, либо регистрации. Окно аутентификации представлена в соответствии с рисунком 2.2.

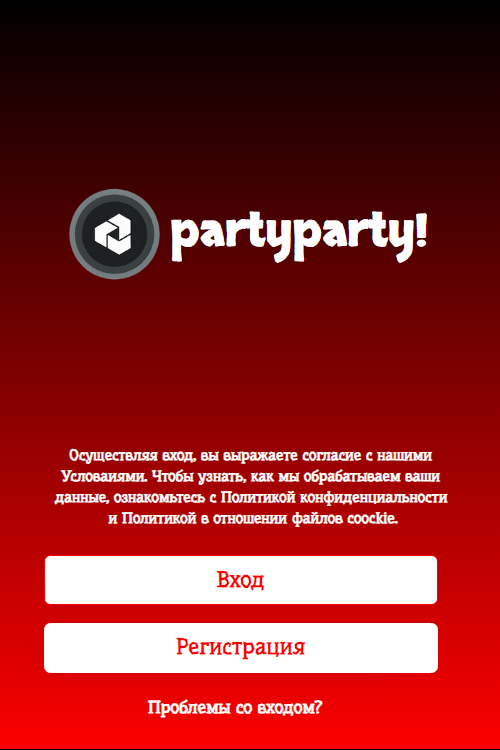


Рисунок 2.2 Окно Аутентификации

При выборе функции регистрации, пользователя перекидывает в новое окно, где ему необходимо внести данные, такие как: имя, логин, пароль и возраст. Окно Регистрации представлена в соответствии с рисунком 2.3.



Рисунок 2.3 Окно Регистрации

При выборе функции Авторизации, пользователя перекидывает в новое окно, где ему необходимо внести данные, такие как: логин и пароль. Окно авторизации представлена в соответствии с рисунком 2.4.

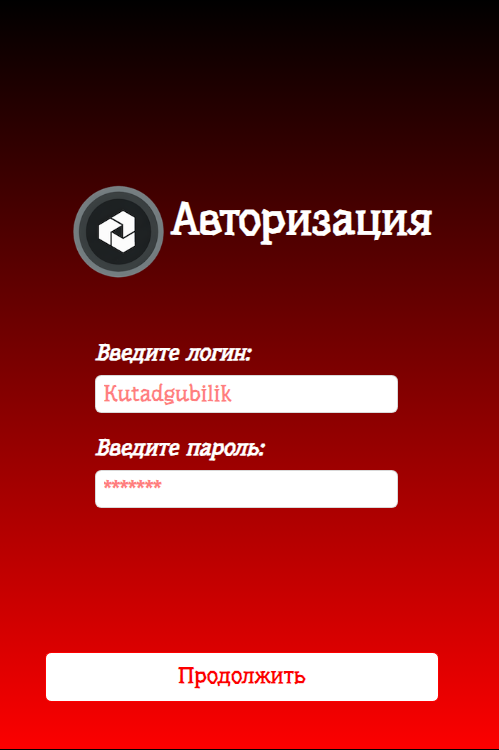


Рисунок 2.4 Окно авторизации

При успешной авторизации, пользователя перекидывает в окно приветствия с правилами пользования приложением. Окно приветствия представлена в соответствии с рисунком 2.5.

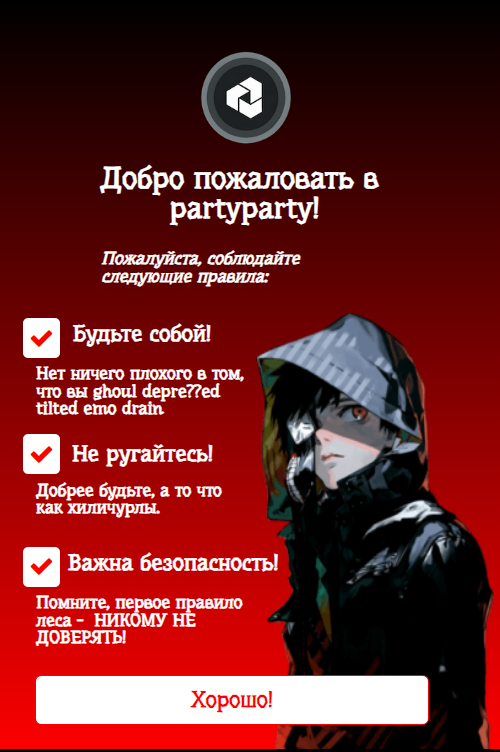


Рисунок 2.5 Приветственное окно

После соглашения с условиями пользования, пользователь переходит в окно с анкетами. Пользователь может либо лайкнуть другого пользователя, либо дизлайкнуть. Окно анкет представлена в соответствии с рисунком 2.6.

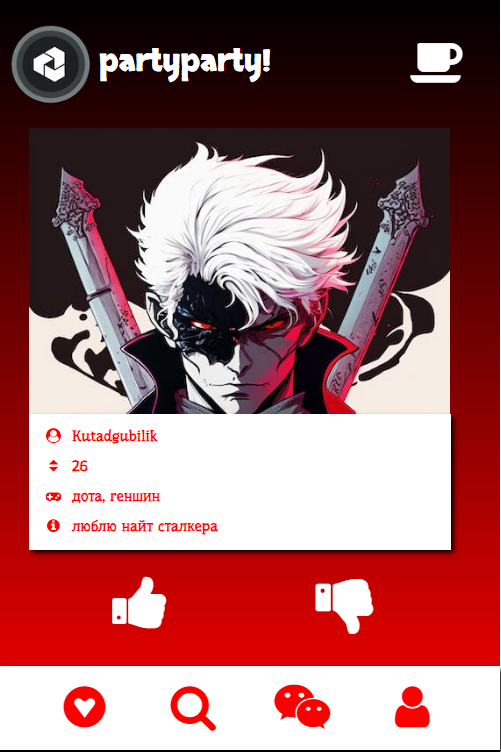


Рисунок 2.6 Окно с анкетами

При нажатии на иконку человека, пользователя перекидывает в окно редактирования анкеты. Пользователь может ввести все необходимые данные о себе, такие как: «О себе», «Steam», «Кого ищу?», «Игры», «Интересы» и «Любимая соц.сеть», а также можно добавить иконку. Окно редактирования анкет представлена в соответствии с рисунком 2.7.



Рисунок 2.7 Окно редактирования анкет

При нажатии на иконку чата, открывается окно чата, где пользователь может общаться с другими пользователями. Окно чата представлена в соответствии с рисунком 2.8.

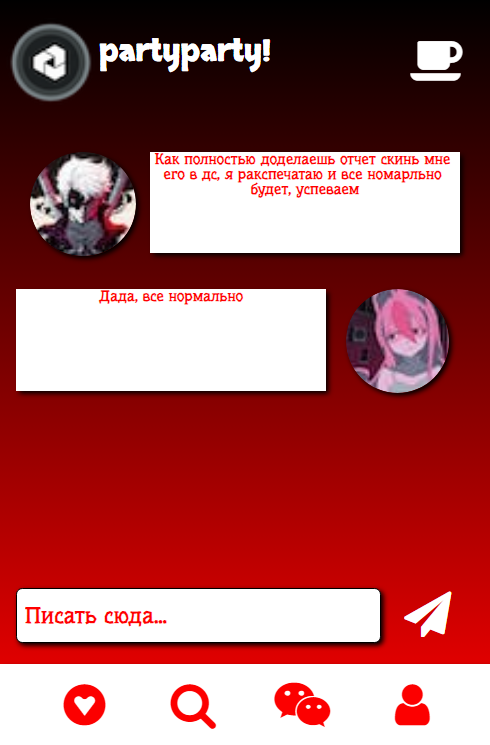


Рисунок 2.8 Окно чата

Проектирование пользовательских интерфейсов является важной задачей в разработке приложений. Цель создания удобного и понятного интерфейса заключается в том, чтобы обеспечить пользователям простой и интуитивно понятный доступ к функциям приложения.

Для достижения этой цели важно следовать некоторым общим принципам проектирования пользовательского интерфейса, таким как использование простых и понятных элементов управления, правильная организация размещения элементов на форме, создание логической структуры приложения и т.д.

При проектировании пользовательских интерфейсов также важно учитывать особенности платформы и целевой аудитории приложения.

В целом, проектирование пользовательских интерфейсов является важным аспектом разработки приложений на этой платформе, и его успешное выполнение может повысить удобство использования приложения для его пользователей и, следовательно, улучшить рейтинг приложения.

# ЗАКЛЮЧЕНИЕ

В результате выполнения данного проекта было разработано no-code приложение под названием «partyparty!», которое предназначено для поиска тиммейтов, которое поможет пользователям быстро и легко найти подходящих тиммейтов на основе их интересов, навыков.

No-code приложение для поиска тиммейтов для совместной игры, разработанное по методологии Scrum, предоставляет уникальную возможность игрокам находить союзников для дальнейшей взаимовыручки. Приложение обладает рядом функций, таких как регистрация, авторизация, изменение профиля, импорт соцсетей, возможность лайкать и дизлайкать чужие анкеты, заходить в профиль других пользователей и функционал чата, которые позволяют игрокам быстро и удобно находить подходящих тиммейтов.

Разработка на платформе no-code значительно сокращает время и затраты на разработку приложения, позволяя сосредоточиться на функционале и пользовательском интерфейсе. Однако, несмотря на это, следует учитывать некоторые ограничения и недостатки, такие как ограниченность функционала и необходимость использования готовых компонентов.

Для реализации проекта были использованы различные методы исследования, включая анализ литературы и изучение аналогов на рынке. Также была проведена работа по разработке no-code приложения, что позволило создать удобный и функциональный интерфейс.

В целом, данное приложение имеет большой потенциал в развитии и может привлечь большое количество пользователей, что может привести к финансовой выгоде для его создателей.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Ушакова, Е.В. Методические указания по выполнению дипломных проектов/работ для бакалавриата направления информационно-коммуникационные технологии [Текст] / Е.В. Ушакова, Л.В. Долматова. -Петропавловск: СКГУ им. М.Козыбаева, 2020. – 99 с.;

2 Методология SCRUM: руководство для разработки продукта в условиях неопределенности / Кен Шваберт, Джефф Сазерленд. - Москва: Питер, 2014. - 496 с.;

3 Agile: гибкие методы разработки программного обеспечения / Кент Бек, Эндрю Хант. - Москва: Издательский дом "Вильямс", 2013. - 352 с.;

4 "Agile: правильное управление изменениями" / Майк Коэн. - Санкт-Петербург: Питер, 2015. - 240 с.;

5 "Разработка программного обеспечения без программирования: no-code платформы и методологии" / Майкл Льюис. - Москва: ДМК Пресс, 2021. - 320 с.;

6 "Agile. Управление проектами и разработка ПО" - Марти Каган. Издательство: БХВ-Петербург, 2017 год.;

7 Семенчук В.: Мобильное приложение как инструмент бизнеса. – Альпина Диджитал, 2016;

8 Колисниченко Д.: Программирование для Android. 3-е издание. – БХВ-Петербург, 2020;

9 Клифтон Я.: Проектирование пользовательского интерфейса Android. 2-е издание. – ДМК-Пресс, 2017;

10 Танг Дж.: Умные мобильные проекты с TensorFlow. – ДМК Пресс, 2018.