Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа № 1. Формирование требований к информационной системе «Мобильное приложение для знакомств»

Студент: Ковкель Н. В.

ФИТ 4 курс 4 группа

Преподаватель: Якубенко К. Д.

Минск 2024

**Введение**

В современном мире, где технологии играют ключевую роль в повседневной жизни, мобильные приложения для знакомств становятся неотъемлемой частью общения и поиска новых связей. Это приложение позволит пользователям легко находить новых знакомых, общаться и выстраивать отношения, будь то для дружбы или романтики. Цель проекта — создать удобную, безопасную и привлекательную платформу, которая облегчит пользователям процесс поиска людей с общими интересами. Среди ключевых задач проекта — разработка интуитивного интерфейса, надежная система обмена сообщениями, алгоритмы интеллектуального подбора пользователей, а также защита личных данных и безопасное взаимодействие. Актуальность проекта обусловлена ростом популярности онлайн-знакомств, особенно среди людей с плотным графиком, которым трудно найти время для новых знакомств в реальной жизни. Приложение будет полезно для людей, стремящихся к новым знакомствам, а также для тех, кто ищет серьезные отношения, предлагая простой и комфортный способ взаимодействия. Целевая аудитория включает пользователей, которые хотят найти партнера для общения, романтических отношений или просто расширить свой круг общения в удобном цифровом формате. Особое внимание будет уделено удобству использования. Приложение будет иметь простой и понятный интерфейс, что позволит пользователям без труда ориентироваться в функционале. Легкость регистрации, интуитивно понятная навигация, возможность моментального начала общения – все эти аспекты станут важными преимуществами продукта. Будут реализованы фильтры поиска, которые позволят настроить параметры поиска по различным критериям: возраст, интересы, местоположение и другим важным факторам. Безопасность и конфиденциальность данных пользователей станут приоритетом. Для этого будет реализована двухфакторная аутентификация, защита данных с использованием современных протоколов шифрования, а также возможность управления приватностью профиля. Это создаст доверительную атмосферу, в которой пользователи смогут чувствовать себя защищенными и уверенными в процессе использования приложения.

1. **Основания для разработки**

Разработка платформы для организации мероприятий и общения между пользователями обусловлена современными тенденциями и потребностями рынка, где технологии играют ключевую роль в создании цифровых сообществ. В условиях активной цифровизации общества, удобство, быстрота взаимодействия и доступность информации становятся важнейшими факторами, обеспечивающими успешное взаимодействие между людьми. Приложения, предлагающие возможность участия в мероприятиях и знакомства с новыми людьми, получают конкурентные преимущества, привлекая пользователей через удобный и интуитивный интерфейс.

Проект реализуется в рамках практической работы над созданием мобильного приложения с функциональностью организации событий, взаимодействия пользователей и обмена сообщениями, что делает его актуальным инструментом для людей с плотным графиком, стремящихся к новым знакомствам или участию в интересных мероприятиях. Приложение предоставляет пользователям возможность создавать события, присоединяться к ним, находить единомышленников и общаться в реальном времени.

Необходимость создания такого решения продиктована растущим спросом на приложения, которые упрощают социальные взаимодействия, особенно среди пользователей, которые ценят гибкость и возможность общаться и участвовать в мероприятиях независимо от времени и места. Приложение помогает оптимизировать процесс поиска людей и событий, улучшает пользовательский опыт и облегчает доступ к мероприятиям, адаптируясь под индивидуальные предпочтения каждого.

В условиях высокой конкуренции в сфере социальных и событийных приложений, создание такой платформы позволит пользователям получать уникальный опыт общения и участия в мероприятиях, а компании — повысить привлекательность своего сервиса и удовлетворить ожидания клиентов.

Таким образом, разработка этой платформы способствует повышению качества социальных взаимодействий, облегчает процесс поиска и участия в событиях, а также помогает пользователям находить новых знакомых. В конечном итоге, проект способствует росту пользовательской базы и увеличению популярности приложения за счет удобства использования и функциональных возможностей.

1. **Назначение разработки**

Целью разработки является создание современного цифрового инструмента, который обеспечивает эффективное и удобное взаимодействие между пользователями и событиями. В условиях активного роста интереса к онлайн-мероприятиям и социальным взаимодействиям, внедрение таких решений позволяет значительно улучшить процесс поиска и участия в событиях, предоставляя пользователям гибкий доступ к информации о мероприятиях и возможностях общения в любое время и из любой точки мира.

Важной особенностью платформы является возможность персонализации взаимодействия, что значительно упрощает процесс нахождения интересных событий и людей для общения. Пользователи могут выбирать мероприятия, которые соответствуют их интересам и потребностям, а также находить других участников, с которыми они могут взаимодействовать. Более того, пользователи получают доступ к детализированной информации о каждом событии, включая описание, дату, время и место, что способствует более осознанному выбору мероприятий.

Особое внимание уделяется управлению процессами общения и организации мероприятий. Пользователи могут легко зарегистрироваться, создавать события и присоединяться к ним. Платформа также уведомляет пользователей о статусе мероприятий, обновлениях и специальных предложениях, что помогает организовать взаимодействие и повысить уровень удовлетворенности.

Таким образом, приложение для организации мероприятий играет ключевую роль в повышении эффективности социальных взаимодействий и уровня удовлетворенности пользователей. С его помощью люди могут оптимизировать свои действия, находить новых знакомых и участвовать в интересных событиях, в то время как организаторы мероприятий получают возможность управлять своими проектами более эффективно. Это решение направлено на создание единой социальной экосистемы, которая объединяет интересы пользователей и организаторов, повышая конкурентоспособность платформы и улучшая опыт взаимодействия.

1. **Требования к программному изделию**
   1. **. Требования к функциональным характеристикам**

Функциональные характеристики приложения отличаются в зависимости от роли пользователя. В приложении доступны следующие роли:

* гость;
* пользователь.

Каждая роль имеет свой набор прав доступа и функциональности, что обеспечивает удобство и безопасность использования системы для разных категорий пользователей.

* + 1. **Требования к клиентской части**

Клиентская часть веб-приложения должна обеспечивать надежную и эффективную работу в любом современном браузере, что позволяет пользователям иметь доступ к приложению в онлайн-режиме, не зависимо от используемого устройства. Это требует, чтобы интерфейс приложения был адаптивным, включая мобильные устройства, что особенно важно в условиях современных технологий и мобильных трендов. Веб-приложение должно быть разработано с использованием актуальных технологий, таких как React Native, что обеспечит высокую производительность и плавность работы.

Интерфейс веб-приложения должен быть тщательно продуманным и интуитивно понятным, чтобы пользователи могли легко ориентироваться в системе и эффективно использовать все предоставленные функции. Интерфейс должен способствовать комфортному взаимодействию, минимизируя необходимость в дополнительных усилиях и упрощая процесс выполнения задач. Также интерфейс обязан обеспечивать следующий функционал:

Функционал для пользователя:

* регистрация;
* авторизация;
* возможность взаимодействовать через свайп-систему;
* редактирование профиля (логин, пароль, аватар, предпочтения);
* получение уведомлений о новых сообщениях;
* общение с другими пользователями через чат.

Функционал для гостя:

* регистрация;
* авторизация;

Основные страницы веб-приложения обеспечивают полное покрытие всех функций, необходимых для эффективного взаимодействия с приложением. Каждая из этих страниц играет ключевую роль в создании удобного и эффективного интерфейса, обеспечивая пользователей всем необходимым для успешного взаимодействия с приложением.

**3.2. Требования к надежности**

Программное средство для выполнения требований надежности должно:

* хранить пароль пользователя в зашифрованном виде;
* ограничить возможность создания более одной учетной записи на одну электронную почту;
* передавать данные по зашифрованному протоколу HTTPS;
* обеспечивать валидацию введенных пользователем данных;
* ограничивать количество запросов по времени для предотвращения DDoS-атак;
* обеспечивать защиту личных данных пользователей;
* обеспечивать идентификацию, аутентификацию и авторизацию пользователей.

**3.3. Условия эксплуатации**

Приложение должно быть спроецировано таким образом, чтобы предоставить возможность для комфортного использования и круглосуточного доступа пользователей. Для достижения этого, интерфейс приложения должен быть интуитивно понятным и удобным в использовании.

**3.4. Требования к составу и параметрам технических средств**

Минимальные аппаратные требования к пользовательской машине:

* процессор: 4-ядерный, 1.5 ГГц или выше;
* минимальный объем оперативной памяти – 2 Гб;
* минимальный доступный объем дискового пространства – 100мб.

Рекомендуемые аппаратные требования к пользовательской машине:

* процессор: 4-ядерный, 1.5 ГГц или выше;
* минимальный объем оперативной памяти – 2 Гб;
* минимальный доступный объем дискового пространства – 100мб.

**3.5. Требования к информационной и программной совместимости**

Язык программирования, используемый серверной частью – Python. В качестве платформы для выполнения серверного кода применяется Django Rest Framework.

Язык программирования, используемый клиентской частью – JavaScript. Клиентская часть разработана с использованием React Native для обеспечения быстрого и отзывчивого пользовательского интерфейса.

Исходный язык, используемый для написания структурированных запросов к базе данных – SQL.

Информационные структуры на входе и выходе для передачи данных между клиентом и сервером – JSON. Передача данных осуществляется по протоколу HTTP.

Для связи между сервером и СУБД. используется база данных PostgreSQL. Это решение позволяет хранить и обрабатывать данные на удаленных серверах, обеспечивая высокую доступность, отказоустойчивость и гибкость масштабирования системы.

Передача данных между клиентской и серверной частями происходит с использованием структурированных данных в формате JSON (JavaScript Object Notation). Это универсальный формат, который легко читается как машинами, так и людьми, обеспечивая быстрый и эффективный обмен информацией. Данные передаются по протоколу HTTP(S), что гарантирует безопасность передачи через зашифрованные соединения.

Также используется WebSocket для реализации некоторых функций в реальном времени, таких как обмен сообщениями в чатах или обновление карты событий, что значительно улучшает взаимодействие с пользователем и позволяет поддерживать высокую степень интерактивности.

Вся система построена с учетом возможности расширения и гибкости, что позволит легко добавлять новые функции или интегрировать сторонние сервисы в будущем, сохраняя при этом высокую производительность и надежность работы приложения.

1. **Требования к программной документации**

Предварительный состав программной документации представляет собой Swagger-документацию, которая используется для описания API веб-приложения. Swagger-документация обеспечивает удобное взаимодействие с программным интерфейсом и помогает разработчикам и пользователям системы легко понять структуру и функциональность API.

Программная документация должна содержать:

* описание пути, метода и названия конечной точки: каждый запрос должен быть четко задокументирован, включая URL-адрес, HTTP-метод (GET, POST, PUT, DELETE и т.д.) и название конечной точки, чтобы разработчики могли правильно обращаться к API;
* пример данных, подаваемых на вход: документация должна содержать примеры данных, которые должны быть отправлены на сервер, включая структуру, типы данных и обязательные поля, чтобы пользователи понимали, какие параметры передавать для корректного запроса.;
* пример данных, возвращаемых с сервера: для каждого запроса должны быть приведены примеры данных, которые возвращаются сервером, включая структуру ответа, возможные поля и их значения. Это помогает пользователям системы понять, какую информацию они получат в результате запроса;
* коды возвращаемых ответов и пояснения к ним: должны быть описаны все возможные коды ответов, которые возвращает сервер, такие как 200 (успешный запрос), 400 (неправильный запрос), 404 (ресурс не найден), 500 (внутренняя ошибка сервера), а также пояснения к каждому коду, чтобы разработчики знали, как интерпретировать результаты;
* дополнительное описание конечной точки, при необходимости: если какая-то конечная точка имеет особенности или дополнительные требования, которые не охвачены стандартными пунктами, они должны быть указаны в этом разделе. Это может включать ограничения по времени обработки запроса, требования к аутентификации или дополнительные параметры, которые могут быть необязательными, но влияют на поведение API.

Кроме того, документация может включать информацию о версиях API, ограничениях на частоту запросов (rate limits), а также советы по оптимальному использованию конечных точек для повышения производительности и безопасности работы системы.

1. **Стадии и этапы разработки**

Разработка должна быть проведена в 3 этапа:

* техническое задание;
* технический проект;
* внедрение.

На стадии «Техническое задание» происходит детальное планирование и формулирование требований к программному продукту. Составляется техническое задание (ТЗ), которое должно включать в себя всю необходимую информацию о целях, назначении, функциях и требованиях к системе. Техническое задание проходит этапы разработки, согласования с заказчиком и утверждения. Этот документ служит основой для всех последующих этапов разработки, поэтому его полнота и точность являются критически важными. Без утвержденного ТЗ невозможен переход к следующему этапу.

На стадии «Технический проект» начинается непосредственная разработка программного обеспечения. Этот этап включает несколько ключевых шагов:

* разработка программы;
* разработка программной документации;
* испытание программы.

На этапе разработки программы программа создается на основе утвержденного технического задания. Здесь выполняется непосредственно программирование, включающее создание кода, интеграцию необходимых модулей и функциональных компонентов. Программирование сопровождается процессами отладки, которые обеспечивают исправление ошибок и корректную работу всех функций. Также на этом этапе могут проводиться предварительные тестирования отдельных модулей.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101--77.

На этапе испытания программы проводится тестирование всей системы, чтобы убедиться в правильной работе всех компонентов и модулей, а также соответствие системы заданным требованиям. Проводятся функциональные, нагрузочные и интеграционные тесты, чтобы проверить работоспособность программы в различных сценариях. Испытания должны выявить и устранить все возможные ошибки перед переходом к следующему этапу.

1. **Описание информационной вербальной модели**

Объекты информационной вербальной модели для приложения для знакомств могут быть представлены следующим образом:

1. Пользователь:
   * описание: представляет собой физическое лицо, использующее платформу для поиска событий и общения. Пользователи могут просматривать события на карте, фильтровать их по критериям, подписываться на других пользователей, отправлять сообщения и участвовать в мероприятиях;
   * атрибуты: имя, фамилия, адрес электронной почты, номер телефона, дата рождения, фотография профиля;
   * **связи:** пользователи связаны с событиями (каждый пользователь может создавать или участвовать в нескольких событиях), с чатом (пользователи могут обмениваться сообщениями), а также с черным списком (пользователь может блокировать других пользователей).
2. Событие:
   * описание: представляет собой мероприятие, которое может быть создано пользователем на платформе. События могут быть различных категорий (социальные, развлекательные, образовательные и т.д.) и содержат информацию о времени и месте проведения;
   * атрибуты: название события, описание, категория, дата и время;
   * **связи:** событие связано с пользователями (пользователи могут создавать или присоединяться к событиям).
3. Чат:
   * описание: средство общения между пользователями платформы. Чат предоставляет возможность отправлять текстовые сообщения и обмениваться данными в рамках взаимодействия участников событий;
   * атрибуты: идентификатор чата, список участников, история сообщений;
   * **связи:** чат связан с пользователями (каждый пользователь может участвовать в нескольких чатах), а также с событиями (чаты могут быть привязаны к конкретным событиям).
4. Подписка:
   * описание: представляет собой связь между пользователями, при которой один пользователь подписывается на обновления другого. Подписка позволяет следить за активностью интересных пользователей, получая уведомления о созданных ими событиях;
   * атрибуты: идентификатор пользователя, на которого подписан, дата подписки;
   * **связи:** подписка связана с пользователями (один пользователь подписывается на другого).
5. **Технико-экономические показатели**

Ориентировочная экономическая эффективность: Разработка онлайн-платформы для организации мероприятий и взаимодействия пользователей обеспечит значительное снижение затрат на физическое присутствие, включая аренду помещений для встреч и оплату услуг по организации событий. Внедрение автоматизированной системы управления мероприятиями и пользователями сократит издержки на координацию событий и повысит скорость обработки заявок. Ожидается, что благодаря удобству использования платформы и расширенному функционалу, объем активности пользователей возрастет на 20-30% в течение первого года после запуска.

Предполагаемая годовая потребность: В первый год работы платформа планирует обслужить до 30 000 мероприятий, что соответствует потребностям как индивидуальных пользователей, так и организаторов. С учетом роста популярности онлайн-взаимодействий, количество мероприятий ожидается на уровне 50 000 в последующие годы. Платформа рассчитана на обработку больших объемов данных и способна масштабироваться в зависимости от увеличения пользовательской активности.

Экономическая выгода проекта заключается также в снижении зависимости от человеческого фактора: автоматизация процессов организации мероприятий и управления пользователями уменьшает количество ошибок, связанных с обработкой данных, и необходимость привлечения большого числа сотрудников. Ожидается, что экономия на операционных расходах составит до 25% по сравнению с традиционными моделями организации событий, что позволит ускорить возврат инвестиций и повысить рентабельность платформы в течение первых двух лет эксплуатации.

По сравнению с отечественными и зарубежными аналогами, такими как Eventbrite и Meetup, предлагаемая платформа отличается большей адаптивностью и настраиваемостью под нужды пользователей. Использование современных технологий, таких как облачные решения для хранения данных и интеграция с системами аналитики, позволяет снизить затраты на техническое обслуживание. За счет автоматизированных процессов организации мероприятий и управления пользователями, уменьшаются временные издержки на выполнение операций, что приводит к снижению операционных затрат на 15-20%. Кроме того, уникальная система фильтрации событий и персонализация предложений повышают уровень удержания пользователей и конверсию в участие, что выгодно отличает платформу от конкурентов.

**Заключение**

Разработанная платформа для организации мероприятий и общения пользователей представляет собой современное и эффективное решение для создания и управления событиями. Она предоставляет пользователям удобный доступ к интерактивной карте событий, возможность фильтрации мероприятий, просмотра профилей других участников, а также участия в чатах. Платформа обеспечивает гибкость и простоту использования как для обычных пользователей, так и для организаторов, которые могут создавать, редактировать и управлять событиями, отслеживая активность и взаимодействие.

Для организаторов платформа предлагает инструменты для создания и редактирования событий, управления списком участников и отправки уведомлений, что значительно упрощает организацию мероприятий. Пользователи получают интуитивно понятный интерфейс для просмотра мероприятий, присоединения к ним, общения с другими участниками и получения уведомлений о новых событиях.

Серверная часть платформы построена на базе технологии Django с использованием фреймворка Django Rest Framework (DRF), что позволило создать высокопроизводительное и масштабируемое решение для обработки пользовательских запросов. Архитектура системы включает роутеры, контроллеры и middleware для обработки данных и поддержания стабильной работы приложения. Взаимодействие между клиентом и сервером осуществляется с помощью REST API, что обеспечивает быструю и надежную передачу данных.

Для обеспечения безопасности данных используется протокол HTTPS, что гарантирует защиту личной информации пользователей и безопасность взаимодействий. Это помогает создать доверительные отношения между участниками платформы и гарантирует защиту данных в процессе использования приложения.

Клиентская часть разработана на основе React Native, что позволило создать интерактивный и удобный мобильный интерфейс для работы с событиями, пользователями и чатами. Использование библиотеки Axios для отправки запросов на сервер оптимизировало работу с данными, а применение Redux для управления состоянием приложения сделало процесс взаимодействия с платформой более плавным и эффективным для пользователей.

В качестве системы управления базами данных выбрана PostgreSQL, обеспечивающая надежное хранение данных о пользователях, событиях и взаимодействиях.