

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

**ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ
ДЕТСКОЙ ЧАСТНОЙ БИБЛИОТЕКИ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы

направления 38.03.05 — Бизнес-информатика

механико-математического факультета

Мартюшенко Ольги Максимовны

Научный руководитель

доцент, к. ф.-м. н.

Р. Н. Фадеев

Заведующий кафедрой

д. ф.-м. н., доцент

С. П. Сидоров

Саратов 2024

ВВЕДЕНИЕ

Актуальность темы. Современные дети увлечены технологиями и мобильными устройствами. Мобильное приложение для библиотеки может быть интересным способом привлечь их внимание к чтению и образовательным материалам. Информационная система может быть мощным инструментом для обучения и развития детей, предоставляя им интерактивный доступ к книгам и образовательным ресурсам.

Потенциал разработки мобильного приложения для детской частной библиотеки весьма значителен. Вот некоторые ключевые аспекты, которые подчеркивают этот потенциал:

- **Инновационное обучение:** Приложение может включать игровые элементы, интерактивные задания и аудиокниги, чтобы сделать процесс обучения более увлекательным.
- **Доступ к редким изданиям:** Через приложение можно предоставить доступ к редким и ценным книгам, которые трудно найти в обычных библиотеках.
- **Социальное взаимодействие:** Функции социального взаимодействия, такие как обмен отзывами и рекомендациями, могут способствовать формированию сообщества юных читателей.
- **Аналитика и отчетность:** Сбор данных о предпочтениях и поведении читателей может помочь библиотеке лучше понять свою аудиторию и адаптировать предложения.
- **Персонализация:** Использование алгоритмов машинного обучения для персонализации рекомендаций книг может значительно улучшить пользовательский опыт.

Целью бакалаврской работы является проектирование мобильного приложения для детской частной библиотеки.

Объектом исследования является само приложение и его пользователи, то есть дети и их родители, которые будут использовать приложение для доступа к книгам и образовательным ресурсам.

Предмет исследования - книги и образовательные ресурсы, их реализация при помощи мобильного приложения.

При проектировании информационной системы для детской частной библиотеки необходимо решить следующие задачи:

1. Изучение основных инструментов системного анализа.
2. Изучение теоретических основ создания мобильных приложений.
3. Планирование проекта.
4. Сбор требований для мобильного приложения и выбор инструментов для создания информационной системы.
5. Описание пользовательских взаимодействий с системой и архитектуры мобильного приложения.
6. Создание прототипов дизайна.
7. Разработка функционала каталога книг с возможностью перехода по ссылкам на сторонний ресурс для ознакомления с информацией о книгах.

Практическая значимость

Практическая значимость исследования мобильного приложения для детской частной библиотеки заключается в том, что оно может принести реальную пользу пользователям и обществу.

Аспекты, которые подчеркивают практическую значимость проекта:

1. Улучшение доступа к образовательным ресурсам: Приложение может предоставить детям удобный доступ к широкому спектру книг и учебных материалов, что особенно важно для тех, кто не имеет легкого доступа к библиотекам.
2. Поддержка образовательного процесса: Интеграция образовательных игр и интерактивных заданий может помочь детям лучше усваивать знания и развивать навыки чтения.
3. Вовлечение родителей: Функции приложения, позволяющие родителям отслеживать прогресс и учебные достижения своих детей, могут способствовать большему вовлечению в образовательный процесс.
4. Безопасность и конфиденциальность: Разработка безопасных механизмов аутентификации и защиты данных повышает доверие пользователей и обеспечивает защиту личной информации.
5. Социальное взаимодействие: Возможность общения и обмена мнениями с другими пользователями через приложение может способствовать

социальному развитию детей.

Данные составляющие не только повышают ценность исследования, но и могут привлечь внимание инвесторов, образовательных учреждений и разработчиков программного обеспечения, заинтересованных в создании полезных и инновационных образовательных инструментов.

Структура и содержание работы. Работа состоит из введения, четырёх разделов, заключения и списка используемых источников, содержащего 20 пунктов.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** раскрывается актуальность темы работы, формулируется тема работы, задачи, которые необходимо решить, и отмечается практическая значимость результатов.

В **первом** разделе описан сбор требований к информационной системе.

Раздел сбора требований к информационной системе — это ключевой элемент работы, который описывает, какие функции и характеристики должна иметь система для удовлетворения потребностей пользователей и достижения целей проекта.

Основные компоненты, которые включены в этот раздел:

1. **Интервью:** Это метод сбора требований, при котором вы ведете беседу с заинтересованными сторонами для выявления их потребностей и ожиданий от системы. Вопросы должны быть структурированы таким образом, чтобы получить как можно больше информации о функциональных и нефункциональных требованиях.
2. **Протокол:** После проведения интервью важно документировать полученную информацию. Протоколы интервью помогают фиксировать ответы участников, их замечания и предложения, что обеспечивает точность и полноту собранных данных.
3. **Техническое задание:** Это документ, который описывает технические и операционные требования системы, а также критерии, по которым система будет разрабатываться и тестироваться. Техническое задание должно быть подробным и четко отражать собранные требования.

4. Трехуровневая шкала приоритетов: Это метод, который помогает определить важность каждого требования. Требования классифицируются как:

- Высокий приоритет: Требования, без которых система не сможет функционировать должным образом.
- Средний приоритет: Требования, которые важны, но не критичны для основных функций системы.
- Низкий приоритет: Требования, которые могут быть добавлены в будущем или не являются необходимыми для основной работы системы.

Во **втором** разделе определены основные составляющие управления проектом:

- Распределение ролей: Объяснение, как роли и ответственности распределяются среди членов команды. В Agile-командах роли могут быть менее формализованы, но каждый член команды должен понимать свои обязанности и то, как его работа влияет на общий проект.
- Agile-разработка: Описание принципов Agile, таких как итеративный подход, адаптивное планирование, ранняя доставка и непрерывное улучшение. Можно также рассмотреть, как Agile-подход влияет на распределение ресурсов и ролей, а также на управление проектом в целом.
- User Story Mapping: Объяснение этого метода как инструмента для планирования и управления разработкой продукта. User Story Mapping помогает команде понять структуру продукта, приоритизировать работу и визуализировать прогресс в контексте пользовательских потребностей.

Третий раздел включает в себя проектирование архитектуры мобильного приложения. В разделе приведен пример Use case "Регистрация в системе мобильного приложения частной библиотеки".

Use case (в переводе с английского - "сценарий использования") - это техника, используемая в разработке программного обеспечения для описания функциональных требований системы. Use case представляет собой описание того, как система будет взаимодействовать с актерами (пользователями или другими системами) для достижения определенных целей. Use case являются важным инструментом для понимания требований системы и обеспечения

соответствия между функциональностью и ожиданиями пользователей. Они также могут быть использованы для создания тестовых сценариев и проверки работоспособности системы.

Также описаны uml-диаграммы, созданные с помощью инструмента PlantUML.

Диаграмма последовательностей (Sequence Diagram) и диаграмма вариантов использования (Use Case Diagram) являются ключевыми элементами в проектировании мобильных приложений. Они помогают визуализировать, как пользователи будут взаимодействовать с приложением, и как система будет реагировать на эти взаимодействия.

Диаграмма последовательностей показывает, как объекты в системе взаимодействуют друг с другом в рамках определенного сценария. Она отображает последовательность сообщений, передаваемых между объектами для выполнения функции или достижения цели.

Диаграмма вариантов использования (Use Case Diagram) отображает функциональные возможности системы с точки зрения пользователей. Она включает актеров (пользователей или другие системы), которые взаимодействуют с системой, и сами варианты использования, представляющие конкретные действия.

Далее представлены low-fidelity и high-fidelity прототипы мобильного приложения.

Low-Fi прототипы — это базовые представления дизайна, которые обычно используются на ранних этапах разработки для общего понимания структуры и основных функций приложения. Они часто создаются вручную или с использованием инструментов, которые не требуют много времени на изучение.

High-Fi прототипы — это более детализированные и интерактивные модели, которые включают в себя дизайн, анимацию и часто интерактивность. Они используются для тестирования пользовательского опыта и получения обратной связи перед финальной разработкой.

С помощью low-fidelity описан процесс авторизации в приложении, учитывая альтернативные потоки. Представлен high-fidelity прототип для основных экранов приложения.

Прототипы созданы с помощью инструментария Figma.

Используя draw io была создана диаграмма классов, содержащая подробное описание классов, их атрибутов и методов, а также отношений между ними.

В **четвертом** разделе описано создание каталога книг с возможностью перехода по ссылкам на сторонний ресурс для ознакомления с описанием. Функционал реализован на Kotlin с Jetpack Compose, REST API и Clean Architecture.

Разработка мобильного приложения на Kotlin с использованием Jetpack Compose, Google API и архитектуры Clean Architecture представляет собой современный подход к созданию структурированных и легко поддерживаемых приложений. Вот краткий обзор того, как эти компоненты могут работать вместе:

Kotlin и Jetpack Compose

Kotlin — это язык программирования, который стал основным для разработки Android-приложений.

Jetpack Compose — это современный инструментарий для создания интерфейса на языке Kotlin, который значительно упрощает разработку мобильных приложений на Android. Он позволяет разработчикам использовать декларативный подход к описанию пользовательского интерфейса, что делает код более чистым и понятным.

Вот краткий обзор ключевых особенностей Jetpack Compose:

- Декларативный UI: Вместо императивного описания интерфейса, как в XML, вы описываете, как UI должен выглядеть в зависимости от состояния приложения.
- Компонентный подход: Интерфейс строится из множества мелких компонентов (Composables), которые можно легко переиспользовать и комбинировать.
- Интеграция с Kotlin: Jetpack Compose полностью написан на Kotlin и хорошо интегрируется с его возможностями, такими как корутины и Flow для асинхронной работы.
- Поддержка тем и стилей: Упрощает работу с темами и стилями, позволяя легко изменять внешний вид приложения.

— Инструменты для разработки: Совместимость с Android Studio и инструментами для профилирования и отладки.

Google API

Использование Google API позволяет интегрировать в приложение множество сервисов Google, таких как карты, авторизация, аналитика и многое другое. Это обогащает функциональность приложения и предоставляет пользователю знакомые сервисы.

Clean Architecture

Clean Architecture — это подход к проектированию приложений, который разделяет код на слои с четко определенными задачами. Это обеспечивает независимость компонентов, упрощает тестирование и поддержку кода.

В **заключении** приведены результаты бакалаврской работы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Описаны требования к будущей информационной системе, создано техническое задание для разработки.
2. Определены основные составляющие управления проектом.
3. Спроектировано взаимодействие пользователя с системой и логика работы приложения.
4. Создан пользовательский интерфейс каталога книг с возможностью перехода по ссылке на сторонний ресурс.
5. Проведен анализ разработанного приложения.