# 1 вариант

**1 слайд цели и задачи**

Целью данного проекта является создании удобного и интуитивно понятного мобильного клиента для пользователей операционной системы Android, который предоставит доступ к информации о Храме Александра Невского.

Для достижения этой цели предстояло выполнить несколько задач. В первую очередь разработать дизайн и интерфейс мобильного клиента, опираясь на веб-сайт храма, чтобы обеспечить пользователей комфортом и легкостью использования приложения.

А также интегрировать функционал для просмотра информации о расписании богослужений, контактной информации и других разделов храма, что позволит пользователям быстро и легко получить необходимую информацию.

Одной из важных задач являлось оптимизация производительности приложения для плавной работы на устройствах под управлением Android. К тому же уделим особое внимание тестированию приложения, чтобы гарантировать его стабильную работу.

**2 слайд актуальность**

Мобильный клиент для Храма Александра Невского является не только уникальным, но и актуальным в контексте поддержки и развития православных ценностей в современном цифровом мире. Приложение значительно улучшит доступность информации о храме для широкой аудитории прихожан Александра Невского с мобильных устройств, способствуя комфорту и привлечению новых посетителей, повышению осведомленности о деятельности храма.

Этот проект демонстрирует использование современных технологий для улучшения взаимодействия организации с пользователем и подтверждает важность цифровизации церковных процессов. Мы надеемся, что данное приложение не только улучшит доступ к информации о храме, но и станет важным шагом в развитии православных ценностей в современном цифровом мире.

**3 слайд native vs web vs cross-platform/multiplatform**

Мобильные приложения имеют ряд преимуществ перед сайтами, адаптированных под мобильные устройства, которые главным образом ориентированы на отображение контента на небольших экранах.

Мобильные клиенты обеспечивают:

Во-первых, высокую скорость работы благодаря кэшированию данных на устройстве. Это позволяет пользователям быстро получать необходимую информацию.

Во-вторых, оффлайн-режим, сохраняя данные и обеспечивая доступ к функционалу даже без интернет-соединения.

Третий плюс - доступ к функциям устройства. Они могут взаимодействовать с камерой, геолокацией, датчиками и другими возможностями устройства, расширяя функциональность приложений.

Кроме того, мобильные приложения предоставляют хороший пользовательский опыт с качественным и интуитивно понятным интерфейсом.

Лучшая возможность для персонализации, оптимизации и адаптации контента под пользователя, а также лучшая производительность и оптимизация для конкретной операционной системы и устройства - все это делает мобильные приложения привлекательным выбором для пользователей.

**4 слайд описание пользовательского интерфейса**

**5 слайд технологии разработки**

В качестве языка программирования решено было использовать Kotlin. Он предлагает современные возможности и инструменты, которые помогут нам создать надежное и эффективное приложение.

Кроме того, был выбран UI-framework Jetpack Compose. Этот инструмент позволит нам создавать красивые и интерактивные пользовательские интерфейсы с помощью декларативного подхода.

Android Studio стала основной средой разработки, так как она предлагает множество полезных функций и интеграцию со всеми необходимыми инструментами для разработки Android-приложений.

Наконец, мы решили использовать систему контроля версий git для отслеживания разработки, изменений и развития данного проекта. Это в перспективе поможет нам эффективно управлять кодом, делать резервные копии и организовать командную разработку.

**6 слайд тестирование**

Мы провели как функциональное, так и нефункциональное ручное тестирование нашего продукта, а также автоматизировали этот процесс с помощью юнит тестов. Целью нашего тестирования было отслеживание возможных ошибок и предотвращение регрессии.

Для более детального анализа работы приложения мы воспользовались логированием и внутренними средствами отладки, что позволило нам быстро выявлять и исправлять проблемы при их возникновении.

Функциональное тестирование помогло нам удостовериться, что наш продукт соответствует спецификациям и требованиям, а нефункциональное тестирование позволило нам проверить производительность, надежность и безопасность нашего продукта.

Особое внимание мы уделили тестированию приложения на различных устройствах с разными версиями операционной системы Android и разрешениями экранов. Это позволило нам гарантировать стабильную работу приложения, а также обеспечить его отзывчивость и качественное функционирование.

**7 слайд архитектура**

Архитектурное решение данного проекта основано на принципах чистой архитектуры, которая предполагает разделение программного обеспечения на три основных слоя: данных (data), бизнес-логики (domain) и представления (presentation). Этот подход позволяет создавать гибкие и масштабируемые программные продукты, где каждый слой отвечает за определенные функциональные возможности и легко может быть заменен или модифицирован без влияния на другие части приложения.

В данном проекте также использован архитектурный паттерн MVVM (Model-View-ViewModel), который специально разработан для создания пользовательских интерфейсов и работает на основе разделения представления данных (Model) от их отображения (View) и логики взаимодействия между ними (ViewModel). Это позволяет создавать удобные и легко поддерживаемые пользовательские интерфейсы, разделяя ответственность между различными компонентами приложения и обеспечивая их независимость.

Такое комбинированное применение принципов чистой архитектуры и паттерна MVVM позволяет создать высококачественное и масштабируемое программное обеспечение, которое легко поддается изменениям и доработкам, сохраняя при этом свою структуру и эффективность работы. Оптимальное сочетание этих подходов обеспечивает удобство в разработке, тестировании и сопровождении проекта, делая его готовым к изменениям и нововведениям в будущем.

# 2 вариант

**1 слайд Цели и задачи:**

- Создать удобный мобильный клиент для пользователей ОС Android, который предоставит доступ к информации о Храме Александра Невского.

- Разработать дизайн и интерфейс, опираясь на веб-сайт храма, для легкого использования приложения.

- Интегрировать функционал для просмотра расписания богослужений, контактов и других разделов храма.

- Оптимизировать производительность для плавной работы на устройствах под управлением Android и провести тестирование для стабильной работы.

**2 слайд Актуальность**

Данный проект является актуальным в контексте поддержки и развития православных ценностей в современном цифровом мире.

Мобильное приложение для Храма Александра Невского сделает информацию о храме доступной широкой аудитории, повысит комфорт посетителей и привлечет новых прихожан, осведомленность о деятельности храма. Это демонстрирует важность цифровизации церковных процессов и улучшения взаимодействия организации с пользователем.

**3 слайд сравнение UI сайта и мобильного клиента**

**4 слайд описание пользовательского интерфейса**

**5 слайд технологии разработки**

Для разработки данного проекта были выбраны следующие технологии:

Язык программирования Kotlin. Он предлагает современные возможности и инструменты, которые помогут нам создать надежное и эффективное приложение.

Jetpack Compose. Этот инструмент предлагает декларативный подход к созданию пользовательского интерфейса, что упрощает процесс разработки, а также он обладает множеством готовых компонентов, которые ускоряют процесс создания интерфейса.

Android Studio. Она стала основной средой разработки, так как она предлагает множество полезных функций и интеграцию со всеми необходимыми инструментами для разработки Android-приложений.

Git. Данная система контроля версий выбрана для отслеживания разработки, изменений и развития данного проекта. Он помогает эффективно управлять кодом, делать резервные копии и организовать командную разработку.

**6 слайд тестирование**

Мы провели как функциональное, так и нефункциональное ручное тестирование нашего продукта, а также автоматизировали этот процесс с помощью юнит тестов. Целью нашего тестирования было отслеживание возможных ошибок и предотвращение регрессии.

Для более детального анализа работы приложения мы воспользовались логированием и внутренними средствами отладки, что позволило нам быстро выявлять и исправлять проблемы при их возникновении.

Функциональное тестирование помогло нам удостовериться, что наш продукт соответствует спецификациям и требованиям, а нефункциональное тестирование позволило нам проверить производительность, надежность и безопасность нашего продукта.

Особое внимание мы уделили тестированию приложения на различных устройствах с разными версиями операционной системы Android и размерами экранов. Это позволило нам гарантировать стабильную работу приложения, а также обеспечить его отзывчивость и качественное функционирование.

**7 слайд архитектура**

Архитектурное решение данного проекта основано на принципах чистой архитектуры, которая предполагает разделение программного обеспечения на три основных слоя: данных (data), бизнес-логики (domain) и представления (presentation).

В данном проекте также использован архитектурный паттерн MVVM (Model-View-ViewModel), который специально разработан для создания пользовательских интерфейсов и работает на основе разделения представления данных от их отображения и логики взаимодействия между ними.

Такое комбинированное применение принципов чистой архитектуры и паттерна MVVM позволяет создать высококачественное и масштабируемое ПО, которое легко поддается изменениям и доработкам, сохраняя при этом свою структуру и эффективность работы. Оптимальное сочетание этих подходов обеспечивает удобство в разработке, тестировании и сопровождении проекта, делая его готовым к изменениям и нововведениям в будущем.