

Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

ОТЧЕТ ПО КУРСОВОЙ РАБОТЕ

«Разработка приложения на Android - Калькулятор»

по дисциплине «Конструирование программного обеспечения»

Выполнил:
студент гр. 5130904/10105

Акрами А.А.

Преподаватель:

Иванов А.С.

Оглавление

<i>Введение</i>	<i>3</i>
<i>Определение проблемы и выработка требований.....</i>	<i>4</i>
<i>Кодирование и отладка</i>	<i>5</i>
<i>Unit тестирование</i>	<i>6</i>
<i>Интеграционное тестирование</i>	<i>7</i>
<i>Разработка архитектуры и детальное проектирование.....</i>	<i>9</i>
<i>Сборка</i>	<i>11</i>
<i>Выводы</i>	<i>12</i>

Введение

Данный учебный проект был разработан с целью ознакомления с основными этапами конструирования программного обеспечения. В качестве целевой платформы была выбрана платформа - Android. Основная же задача проекта заключалась в создании учебного приложения, выполняющего функции базового и научного калькулятора, а также конвертера единиц измерения.

Процесс разработки прошёл через следующие этапы:

1. **Определение проблемы:** Анализ потребностей пользователей и выявление основных задач, которые должно решать приложение.
2. **Выработка требований:** Определение функциональных и нефункциональных требований к приложению, включая функции базового и научного калькуляторов, конвертера единиц измерения и функцию истории вычислений.
3. **Разработка архитектуры и детальное проектирование:** Проектирование структуры приложения и его компонентов, включая создание диаграммы контекста системы и контейнеров.
4. **Кодирование и отладка:** Реализация функциональности приложения с последующей отладкой и исправлением ошибок.
5. **Unit тестирование:** Создание и выполнение юнит-тестов для проверки отдельных компонентов и функций приложения.
6. **Интеграционное тестирование:** Проведение тестов на взаимодействие между различными компонентами приложения, а также проверка работы приложения в целом.
7. **Сборка:** Сборка и подготовка приложения к Beta-тестированию, включая компиляцию и запуск.

В этом отчёте мы рассмотрим каждый из этих этапов подробнее и проанализируем результаты выполненной работы.

Определение проблемы и выработка требований

Современные пользователи нуждаются в универсальном инструменте, который объединяет основные функции вычислений в одном приложении. Раздельные приложения для базовых и научных калькуляций, а также для конвертации единиц измерения, могут быть неудобны и занимать много места на устройстве. Данный проект решает эту проблему, предоставляя единое, удобное и простое в использовании Android-приложение, которое включает в себя все эти функции.

Для достижения целей проекта были реализованы следующие функции:

- **Базовый калькулятор:** поддерживает основные арифметические операции (сложение, вычитание, умножение, деление).
- **Научный калькулятор:** включает расширенные математические функции, такие как тригонометрические, логарифмические и экспоненциальные вычисления.
- **Конвертер единиц измерения:** позволяет конвертировать различные единицы измерения, такие как длина, масса, объем, температура и другие.
- **История вычислений:** сохраняет и отображает предыдущие вычисления для удобства пользователя.

Кодирование и отладка

Процесс разработки приложения включал следующие этапы кодирования и отладки:

- **Проектирование архитектуры:** Создание структуры проекта, определение основных компонентов и их взаимодействий. Использование архитектурного подхода MVVM (Model-View-ViewModel) для улучшения поддерживаемости и тестируемости кода.
- **Разработка пользовательского интерфейса (UI):**
 - Создание удобного и интуитивно понятного интерфейса для базового и научного калькулятора, а также конвертера единиц измерения.
 - Обеспечение адаптивности интерфейса для различных размеров экранов и ориентаций устройств.
- **Реализация функциональности:**
 - Разработка основной логики для выполнения арифметических и научных вычислений.
 - Реализация конвертера единиц измерения с поддержкой различных типов конвертаций.
 - Создание механизма для сохранения и отображения истории вычислений.
- **Отладка и тестирование:**
 - Проведение отладки кода с использованием встроенных инструментов Android Studio.
 - Написание и выполнение unit тестов для проверки корректности работы всех основных функций.
 - Интеграционное тестирование для проверки взаимодействия различных компонентов приложения.
- **Оптимизация и улучшение производительности:**
 - Оптимизация алгоритмов вычислений для повышения скорости и снижения потребления ресурсов.
 - Обеспечение плавности и отзывчивости пользовательского интерфейса.
- **Документирование кода:**
 - Добавление комментариев и документации к ключевым частям кода для облегчения понимания и поддержки в будущем.

Unit тестирование

Для обеспечения качества и надежности приложения были разработаны и внедрены следующие **unit тесты**:

- **Тестирование базового калькулятора:**
 - Проверка корректности выполнения основных арифметических операций (сложение, вычитание, умножение, деление).
 - Обработка деления на ноль и других исключительных ситуаций.
- **Тестирование научного калькулятора:**
 - Проверка точности тригонометрических вычислений (синус, косинус, тангенс и т.д.).
 - Тестирование логарифмических и экспоненциальных функций.
 - Валидация ввода и обработка ошибок.
- **Тестирование конвертера единиц измерения:**
 - Проверка корректности конвертации различных единиц измерения (длина, масса, объем, температура и т.д.).
 - Обработка некорректных данных и исключений.
- **Тестирование истории вычислений:**
 - Проверка сохранения и отображения истории вычислений.
 - Тестирование корректного удаления и очистки истории.

Все тесты написаны с использованием стандартных инструментов для тестирования Android приложений, таких как JUnit и обеспечивают полное покрытие критических функциональных частей приложения.

Интеграционное тестирование

Для обеспечения корректного взаимодействия различных компонентов приложения были разработаны и выполнены следующие сценарии **интеграционного тестирования**:

Сценарий 1: Интеграция базового калькулятора и истории вычисления

1. Описание:

- Проверка взаимодействия между базовым калькулятором и функцией сохранения истории вычислений.

2. Шаги тестирования:

- Открыть приложение и перейти к базовому калькулятору.
- Выполнить несколько арифметических операций (например, $2 + 3$, $5 * 6$).
- Перейти к истории вычислений.
- Убедиться, что все выполненные операции корректно отображаются в истории.
- Удалить одну запись из истории и проверить, что оставшиеся записи остались без изменений.
- Вернуться к базовому калькулятору и выполнить новую операцию.
- Проверить, что новая операция добавлена в историю.

3. Ожидаемый результат:

- Все выполненные операции должны выполняться правильно и отображаться в истории вычислений.
- Удаление записи из истории должно корректно обновлять список без затрагивания других записей.
- Новые операции должны добавляться в историю корректно.

Сценарий 2: Интеграция научного калькулятора и истории вычисления

1. Описание:

- Проверка взаимодействия между научным калькулятором и функцией сохранения истории вычислений.

2. Шаги тестирования:

- Открыть приложение и перейти к научному калькулятору.
- Выполнить вычисление с использованием тригонометрической функции (например, $\sin(30)$).
- Перейти к истории вычислений.
- Убедиться, что все выполненные операции корректно отображаются в истории.

- Удалить одну запись из истории и проверить, что оставшиеся записи остались без изменений.
- Вернуться к базовому калькулятору и выполнить новую операцию.
- Проверить, что новая операция добавлена в историю.

3. Ожидаемый результат:

- Все выполненные операции должны выполняться правильно и отображаться в истории вычислений.
- Удаление записи из истории должно корректно обновлять список без затрагивания других записей.
- Новые операции должны добавляться в историю корректно.

Сценарий 3: Интеграция конвертера единиц измерения

1. Описание:

- Проверка исправной работы конвертера единиц измерения.

2. Шаги тестирования:

- Открыть приложение и перейти к конвертеру ед. измерения.
- Выбрать тип конвертации (например, площадь, длина, вес, градусы).
- Выбрать единицы измерения (например, из какой условной единицы в какую желаем перевести значение).
- Ввести значение и выполнить конвертацию.
- Проверить результат.

3. Ожидаемый результат:

- Конвертация должна выполняться правильно, и результат должен соответствовать ожидаемому значению.

Эти сценарии помогают удостовериться, что различные функции приложения взаимодействуют друг с другом корректно и обеспечивают ожидаемую функциональность.

Разработка архитектуры и детальное проектирование

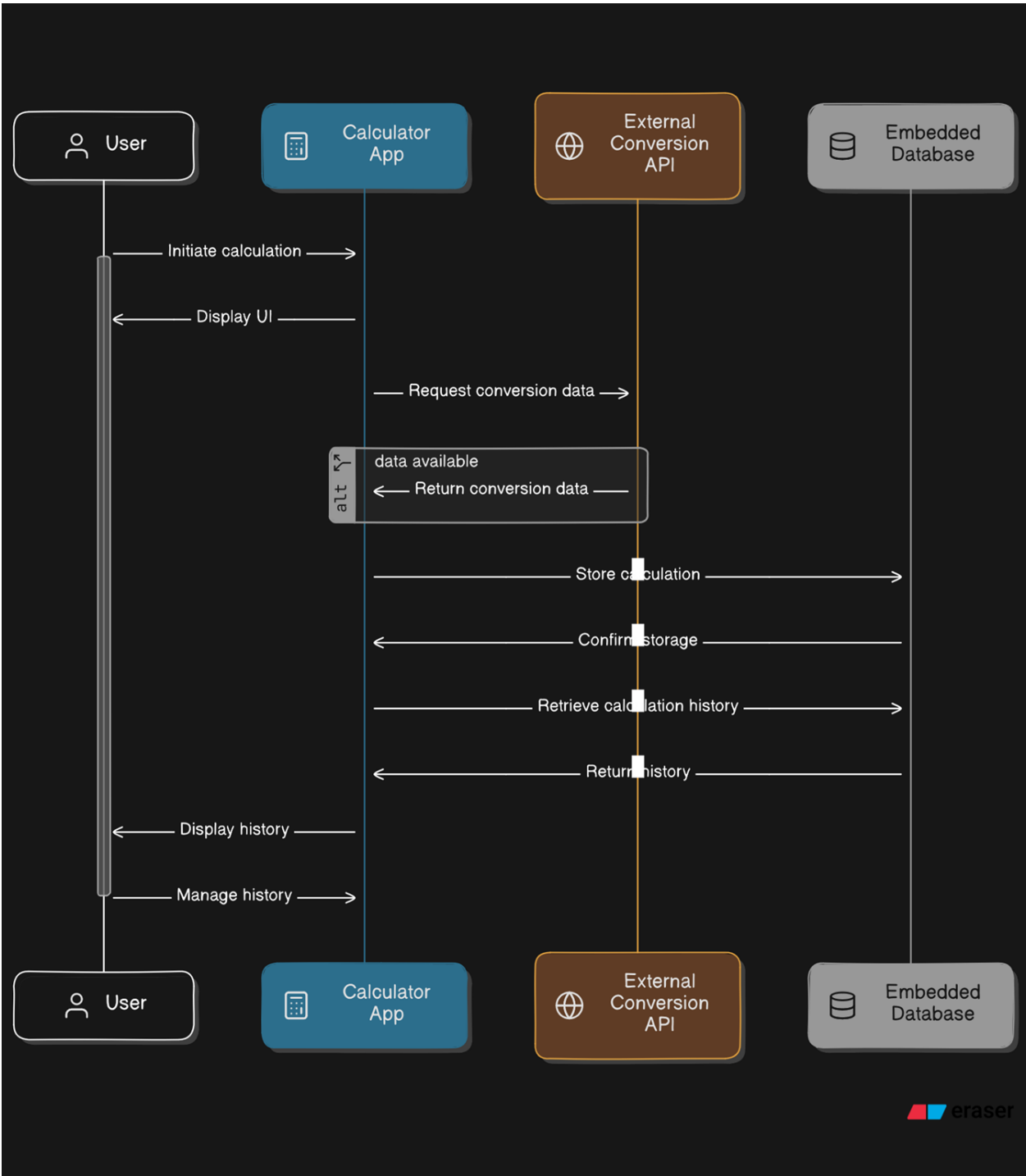


Illustration 1: System Context Diagram

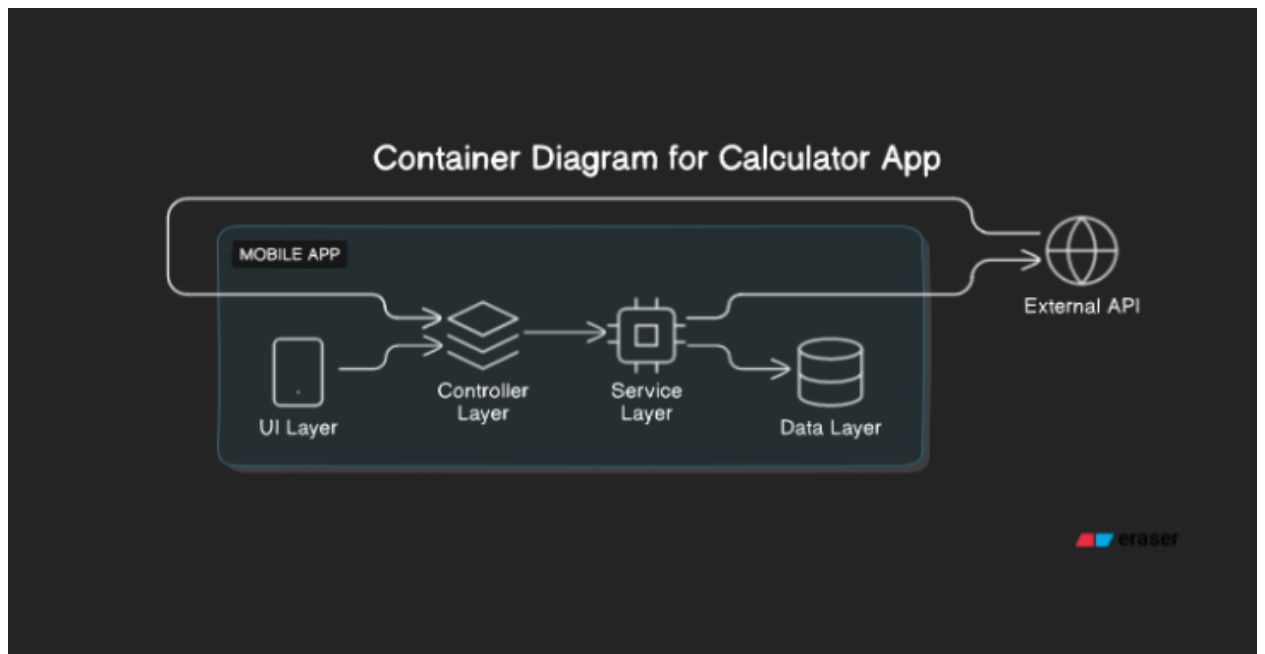


Illustration 2: Container Diagram

Сборка

Цель этапа сборки заключалась в подготовке приложения к бета-тестированию путём компиляции и запуска приложения в контролируемой среде.

1. **Компиляция и сборка кода:** Используя интегрированную среду разработки (IDE) Android Studio, производилась компиляция и сборка исходного кода проекта. В этот процесс входило проверка наличия ошибок и исправление возможных проблем.
2. **Запуск приложения в контролируемой среде:** Собранное приложение запускалось и тестировалось на различных устройствах или эмуляторах Android. В процессе тестирования проводилась проверка функциональности и стабильности работы приложения.
3. **Подготовка к бета-тестированию:** После успешного запуска и тестирования приложения, оно готовилось к дальнейшему бета-тестированию. Это включало подготовку дистрибутивов приложения для загрузки на тестовые устройства или платформы для бета-тестирования.

Этот этап сборки позволял убедиться в работоспособности приложения и готовности его к бета-тестированию перед публикацией.

Выводы

В рамках данного проекта было проведено полное циклическое проектирование и разработка учебного приложения для платформы Android, представляющего собой калькулятор с функциями базового и научного калькуляторов, а также конвертера единиц измерения.

1. **Достижение поставленных целей:** Проект позволил ознакомиться с основными этапами разработки мобильных приложений, начиная от определения проблемы и выработки требований, и заканчивая сборкой и подготовкой приложения к бета-тестированию. Все поставленные цели были успешно достигнуты.
2. **Полученный опыт и уроки:** Работа над проектом позволила углубить знания в области разработки Android-приложений, познакомиться с использованием инструментов и технологий, а также научиться применять лучшие практики разработки и тестирования приложений.
3. **Дальнейшие перспективы:** В процессе разработки были выявлены некоторые возможности для улучшения и доработки приложения. Дальнейшее развитие проекта может включать в себя добавление новых функций, улучшение пользовательского интерфейса, оптимизацию производительности и расширение покрытия тестами.

В целом, проект позволил не только приобрести новые знания и навыки в области мобильной разработки, но и оценить важность систематического подхода к проектированию и разработке программного обеспечения.