



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
филиал ФГБОУ ВО «РГГМУ» в г. Туапсе

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Разработка мобильного приложения по изучению корейского языка
«Korean Simple»»

Исполнитель Земляков Данил Евгеньевич

Руководитель к.т.н., Сафонова Татьяна Владимировна

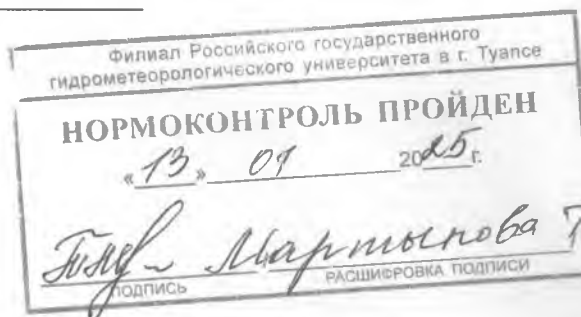
«К защите допускаю»

Руководитель кафедры 

кандидат экономических наук

Майборода Евгений Викторович

«20» 01 2025 г.



Туапсе
2025

ОГЛАВЛЕНИЕ

Введение.....	4
1 Теоретическая часть.....	6
1.1 Анализ предметной области	6
1.2 Анализ среды и выбор методов решения задач.....	12
1.3 Выбор CASE-средств и их обоснование.....	15
2 Проектирование приложения.....	19
2.1 Язык моделирования UML.....	19
2.2 Диаграмма вариантов использования (USE-CASE).....	19
2.3 Сценарии на основе USE-CASE диаграммы.....	20
2.4 Диаграмма последовательности	22
2.5 Диаграмма классов.....	23
2.6 Диаграмма компонентов	24
2.7 Диаграмма развертывания	24
2.8 Модель архитектуры C4	25
2.9 Декомпозиция приложения.....	28
3 Разработка приложения.....	30
3.1 Описание выбранных средств разработки и языка программирования	30
3.2 Описание основных алгоритмов приложения	31
3.3 Описание программных модулей.....	35
3.4 Описание интерфейса приложения	38
Заключение	46
Список литературы	49
Приложение	54

Введение

В последние годы интерес к изучению корейского языка стремительно растёт по всему миру. Это связано с несколькими факторами, включая популяризацию корейской культуры через музыку (K-pop), киноиндустрию, сериалы (дорамы), а также значительное влияние Республики Корея в сфере технологий, экономики и международных отношений. Корейский язык становится не только средством общения, но и способом приобщения к богатой и самобытной культуре Кореи.

Тем не менее, изучение корейского языка остаётся непростой задачей для большинства новичков. Сложности часто возникают из-за уникальности корейской письменности (хангыль), специфических грамматических структур и недостатка качественных и доступных образовательных ресурсов. Традиционные методы обучения, такие как книги или групповые занятия, не всегда обеспечивают достаточную гибкость и интерактивность, что снижает мотивацию учащихся.

На этом фоне мобильные приложения становятся эффективным инструментом для самостоятельного изучения языка. Они позволяют учиться в удобное время и темпе, совмещая обучение с повседневной жизнью. Однако далеко не все существующие приложения могут похвастаться доступностью, простотой использования и полноценным набором функций, необходимых для начинающих.

Мобильное приложение «Korean Simple» разработано с целью преодолеть эти трудности и предложить удобное, интуитивно понятное решение для тех, кто хочет изучать корейский язык с нуля.

Актуальность исследования- основное внимание в приложении уделено созданию сбалансированной системы обучения, которая объединяет интерактивные уроки, проверку знаний и словарный модуль, что помогает пользователям не только осваивать базовые элементы языка, но и закреплять их на практике.

Объект исследования- мобильное приложение «Korean Simple»

Предмет исследования - разработка приложения

Основные задачи, решаемые в рамках данной работы:

- проанализировать выделенную проблему;
- выделить целевую аудиторию и ее потребности;
- спроектировать информационную систему;
- разработать мобильное приложение, которое позволит пользователям не только осваивать базовые элементы языка, но и закреплять их на практике.

В рамках данного проекта акцент сделан на минималистичный дизайн, который не отвлекает от обучения, а также на интуитивную навигацию, позволяющую пользователям с любым уровнем технической подготовки легко освоить приложение. Геймификация, встроенная в функционал, поддерживает мотивацию, превращая процесс обучения в увлекательное занятие.

Настоящая пояснительная записка посвящена описанию этапов разработки, обоснованию выбранных технологий, анализу целевой аудитории и перспективам развития «Korean Simple». Приложение направлено на то, чтобы стать универсальным инструментом для всех, кто интересуется изучением корейского языка, и внести свой вклад в популяризацию корейской культуры на международной арене.

Разработка мобильного приложения «Korean Simple» является важным шагом в направлении упрощения и популяризации изучения корейского языка. Приложение нацелено на то, чтобы стать незаменимым помощником для каждого, кто хочет освоить язык и приблизиться к культуре Кореи.

1 Теоретическая часть

1.1 Анализ предметной области

Мобильное приложение «Korean Simple» предоставляет пользователям возможность изучать корейский язык с помощью различных интерактивных уроков и заданий. Предметная область этого приложения включает в себя изучение основных грамматических правил, лексики, алфавита и произношения корейского языка.

Анализируя предметную область приложения, можно отметить следующие ключевые особенности:

1. Удобство и доступность изучения: приложение предоставляет пользователю возможность изучать корейский язык в любом удобном месте и времени, благодаря доступности на мобильных устройствах.

2. Структурированные уроки: уроки в приложении разделены на различные уровни сложности, что позволяет пользователям начать изучение с самого начала или продолжить обучение на более продвинутом уровне.

3. Интерактивные задания: приложение предлагает разнообразные задания, такие как тесты, игры, аудио- и видеоматериалы, которые помогают улучшить навыки понимания и произношения корейского языка.

4. Постоянное обновление материалов: разработчики приложения постоянно добавляют новые уроки и материалы для обучения, что обеспечивает пользователям возможность расширять свои знания и навыки.

В целом, анализ предметной области мобильного приложения «Korean Simple» показывает, что оно предоставляет широкие возможности для изучения корейского языка и помогает пользователям достичь желаемого уровня владения им.

Мобильное приложение «Korean Simple» создается для людей, которые хотят изучать корейский язык. Цель приложения заключается в том, чтобы предоставить пользователям удобный и эффективный способ освоения нового языка.

Приложение предлагает разнообразные уроки, упражнения, словарь, аудио и видео материалы для изучения корейского языка. Оно помогает пользователям расширить свой словарный запас, изучить правила грамматики, научиться произносить слова правильно и понимать устную и письменную речь на корейском языке.

«Korean Simple» подходит как для начинающих, так и для тех, кто уже изучает корейский язык и хочет улучшить свои знания. Приложение также удобно для людей, которые заняты и не имеют возможности посещать курсы или иметь регулярные занятия с репетитором.

Таким образом, мобильное приложение «Korean Simple» создается для всех, кто интересуется изучением корейского языка и стремится улучшить свои навыки в этой области.

Разработка мобильного приложения по изучению корейского языка обусловлена рядом ключевых факторов, подчеркивающих его актуальность и необходимость в современных условиях:

1. Растущая популярность корейской культуры: Корейский язык становится все более популярным из-за растущего влияния корейской культуры, такой как музыка К-поп [14], драмы К-drama, фильмы и кулинария. Многие люди хотят научиться говорить на корейском, чтобы лучше понимать и наслаждаться этой культурой.

2. Рост туризма: В последние годы число туристов, посещающих Южную Корею, значительно возросло. Путешественники хотят иметь базовые знания корейского языка, чтобы лучше общаться с местными жителями и насладиться своим путешествием.

3. Возможности обучения в любое время и в любом месте: Мобильные приложения обеспечивают доступ к обучению корейскому языку в любое время и в любом месте, что делает их удобными для занятых людей, которые не имеют возможности посещать курсы или уроки.

4. Интерактивное обучение: Мобильные приложения могут предлагать интерактивные методики обучения, такие как игры, тесты, аудио- и

видеоуроки, что делает изучение корейского языка более увлекательным и эффективным.

5. Персонализированный подход: Мобильные приложения могут предлагать индивидуальные уроки и задания, учитывая уровень знаний и потребности каждого пользователя, что помогает им быстрее и эффективнее усваивать материал.

Ключевыми модулями приложения «Korean Simple» являются:

- модуль «Уроки» – предоставляет список уроков посвящённых обучению иностранному языку;
- модуль «Тесты» – представляет возможность пройти тест на знание корейского языка;
- модуль «Словарь» – содержит сборник популярных необходимых слов с переводом на русский язык;

Предполагаемый функционал представлен в таблице 1 с распределением на категории «необходимых», «желательных», «возможных» и «отсутствующих».

Таблица 1- Функционал разрабатываемой программы

Необходимые	Желательные	Возможные	Отсутствующие
Предоставление учебных материалов на тему иностранного языка	Заголовок категории типа отходов, пример отходов, описание	Интеграция с внешней базой данных уроков	Интеграция с AR для идентификации предметов вокруг
Возможность пройти экзамен на знание иностранного языка	Возможность увидеть неправильные ответы	Система достижений в обучении	Подбор случайных вопросов по теме с использованием ИИ
Возможность изучить иностранные слова по отдельности	Перевод слов с русского на корейский	Интеграция с внешними словарями	Возможность создания своего словаря с переводами

Опишем функциональные и нефункциональные требования разрабатываемого проекта.

Функциональные требования:

- приложение обязано иметь только достоверную информацию;
- приложение должно иметь доказательства корректного обучения иностранному языку;
- тест должен корректно оценивать знания пользователя в тесте;
- приложение должно быть удобным и интуитивно понятным в использовании.

Нефункциональные требования:

- приложение должно быстро работать и быть стабильным;
- приложение должно иметь привлекательный интерфейс;
- приложение должно быть доступно на Android;
- приложение должно быть на русском языке.

Анализ доступных решений показывает, что существует много платформ, предлагающих разные методы изучения иностранного языка. На рынке цифровых продуктов существует ряд решений, связанных с изучением иностранного языка.



Рисунок 1 - Приложение Simpler

Приложение «Simpleg» [1] отлично адаптировано для изучения языка всей семьей благодаря возможности совершенствовать навыки в игровой форме. Особенность Simpleg являются домашние задания в формате детективных историй. Пример интерфейса приложения изображен на рисунке 1.

В приложение «Puzzle English» [2] для обучения задействованы традиционные уроки, аудио-пазлы, видео-пазлы, мини-игры, подкастыи еще много всякого. Уникальность приложения«Puzzle English»в возможности обучаться английскому языку по фильмам и сериалам.

Пример интерфейса приложения изображен на рисунке 2.

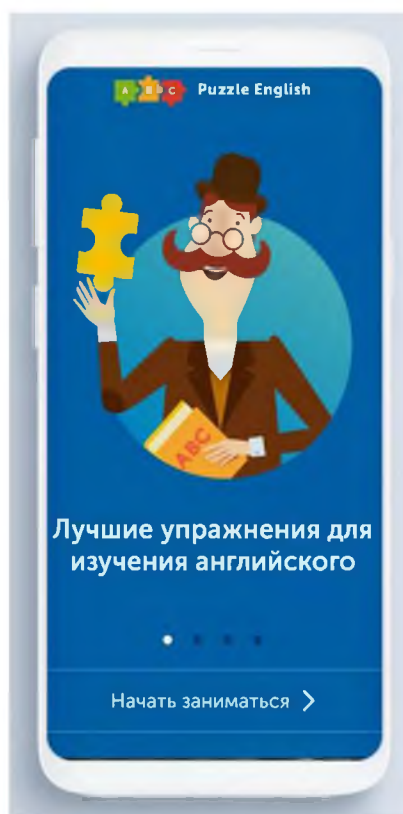


Рисунок 2 - ПриложениеPuzzle English

Приложение «LingoDeer» [3] – фокусируется на обучении азиатских языков, но английский язык тоже присутствует.В основу LingoDeer заложены классические методы — флэш-карты, тесты, статистика достижений в изучении языка. После завершения первого урока дальнейшее обучение доступно лишь после переключения на платную версию аккаунта.

Пример интерфейса приложения изображен на рисунке 3.



Рисунок 3 - Приложение LingoDeer

Сравнение основных характеристик аналогов приложений и своего разработанного приложения «Korean Simple» по основным функциональным характеристикам представлено в таблице 2.

Таблица 2 - Сравнение косвенных аналогов по функционалу

	Korean Simple	Simple	Puzzle English	LingoDeer
Язык	Kotlin	Java/Kotlin	React Native	React Native
Функционал	Уроки, тест, словарь	Уроки с домашними заданиями и словарь	Видеоуроки, аудиоуроки	Тесты, словарь
Стоимость	Бесплатно	Платное с дополнительными функциями	Платное с расширенными функциями	Платное с базовыми функциями и подписками

Продолжение таблицы 2

Надёжность	Высокая, стабильная работа, регулярное тестирование	Низкая, частые сбои, медленная работа	Низкая, нестабильная работа приложения	Средняя, нерегулярное обновление
Качество	Высокое, удобный и интуитивный интерфейс, точная и актуальная информация	Низкое, неудобный интерфейс, неточная информация	Среднее, недостаточно информации, неинтуитивный интерфейс	Среднее, отсутствуют некоторые важные функции

Основное преимущество приложения «Korean Simple» заключается в его комплексном подходе. В отличие от аналогов, ограничивающихся одним или двумя функциональными модулями, приложение объединяет в себе как модуль обучения так и модуль проверки знаний. Это обеспечивает более полное покрытие знаний пользователей и способствует более эффективному формированию знаний корейского языка.

Приложение «Korean Simple» отличается интеграцией различных функциональных модулей в единую систему. Это позволяет пользователю получить полную картину и использовать все инструменты для улучшения своих навыков в области иностранного языка. Другой важной особенностью является удобный и интуитивно понятный интерфейс, направленный на максимальное упрощение взаимодействия с приложением.

1.2 Анализ среды и выбор методов решения задач

В рамках проекта мобильного приложения «Korean Simple» важно определить и чётко структурировать подходы к решению поставленных задач.

На начальном этапе разработки неизбежно возникает определённая неопределённость, связанная с переходом от общей концепции к конкретным условиям её реализации и достижению конечной цели. Для устранения этой неопределённости необходимо сформулировать гипотезу, которая станет основой для построения модели реализации.

Формулирование гипотезы потребует анализа текущих проблем целевой аудитории, формулировки ключевых требований к функционалу приложения и определения основных методов их реализации. На её основе будут разрабатываться ключевые компоненты системы: структура пользовательского интерфейса, алгоритмы взаимодействия пользователя с обучающим контентом, а также подходы к мотивации и оценке прогресса.

В процессе построения модели реализации особое внимание будет уделено адаптивности и гибкости разрабатываемой системы. Это обеспечит возможность дальнейшего масштабирования приложения, добавления новых функций и модулей без существенного изменения основной архитектуры. Кроме того, модель позволит своевременно вносить корректировки на основе обратной связи от пользователей и результатов тестирования.

Таким образом, чёткая структуризация подходов, основанная на сформулированной гипотезе, станет фундаментом успешной реализации проекта. Она создаст необходимую основу для последовательного выполнения этапов разработки, что в конечном итоге обеспечит высокое качество продукта и соответствие его функционала ожиданиям пользователей.

Выбор методик разработки будет обусловлен следующими факторами:

- сложность поставленных задач;
- интеграция с существующей инфраструктурой;
- ограничение по ресурсам;
- удобство использования приложения.

Для эффективного решения задач разработки применены следующие методы:

- метод структурированного анализа и проектирования;

- метод объектно-ориентированного программирования;
- метод тестирования и отладки;
- метод управления рисками.

Методы планирования представляют собой совокупность инструментов и подходов, используемых для сбора, анализа и обработки информации с целью разработки обоснованных и эффективных планов. Эти методы позволяют структурировать процесс планирования, обеспечивая логичное и последовательное распределение ресурсов, формулирование целей и определение путей их достижения.

Эффективное использование методов планирования способствует оптимизации работы компании или её отдельных подразделений, охватывая такие аспекты, как стратегическое развитие, управление проектами, распределение задач, контроль выполнения и анализ результатов. Они помогают адаптироваться к изменяющимся условиям внешней среды, минимизировать риски и принимать взвешенные управленческие решения.

Применение методов планирования актуально как на уровне долгосрочных стратегических решений, так и для оперативного управления. Их использование позволяет обеспечить системность и прозрачность процессов, способствуя достижению устойчивого развития компании.

Диаграмма Ганта[4] отображает план проекта, показывая этапы работы и сроки их выполнения. Такая диаграмма изображена на таблице 3 со всеми фазами разработки.

Таблица 3 - Диаграмма Ганта

Наименование	Месяц															
	Сен	Ок	Ноя	Дек	Ян	Фе	Ма	Ап	Ма	Ию	Ию	Ав	Се	Ок	Ноя	Де
	т				в	в	рт	р	й	н	л	г	н	т		к
Фаза анализа и планирования требований																
Фаза проектирования																

Продолжение таблицы 3

Фаза построения																	
Фаза тестирования																	
Фаза внедрения																	
Год	2024					2025											

1.3 Выбор CASE-средств и их обоснование

Разработка приложения «Korean Simple» требует CASE-технологии, включающие анализ требований, проектирование, разработку кода, тестирование, документирование и управление проектом. В процессе разработки применяются:

- средства разработки приложений (Android Studio) [18];
- средства планирования и управления проектом (MicrosoftProject).

Для разработки системы используется техническое оборудование с техническими характеристиками:

- тип: персональный компьютер (ноутбук);
- операционная система: Windows 10;
- процессор: Intel Celeron или Ryzen 3;
- оперативная память: 4 ГБ;
- место на диске: 1 ГБ;
- зависимости: Kotlin[19] версии 3.5+;
- браузер: Yandex.

SWOT-анализ служит основой для разработки стратегии развития приложения «Korean Simple». Он помогает идентифицировать сильные и слабые стороны, а также оценить возможности и угрозы, с которыми придется столкнуться.

В таблице 4 представлен SWOT-анализ.

Таблица 4 - SWOT-анализ

SWOT-анализ разрабатываемой системы	
S	<ol style="list-style-type: none"> 1. Актуальность темы. 2. Комплексный подход. 3. Удобство использования. 4. Потенциал для расширения.
W	<ol style="list-style-type: none"> 1. Зависимость от географического покрытия. 2. Зависимость от актуальности данных. 3. Конкуренция.
O	<ol style="list-style-type: none"> 1. Локализация. 2. Расширение функционала. 3. Маркетинг и продвижение.
T	<ol style="list-style-type: none"> 1. Изменение законодательства. 2. Недостаток финансирования. 3. Развитие технологий.

В заключение SWOT-анализа приложения «Korean Simple» можно сказать, что проект имеет значительный потенциал для успеха, опираясь на актуальность темы и комплексный подход к решению проблемы утилизации отходов.

Сильными сторонами проекта являются комплексный функционал приложения, актуальность выбранной темы и значительный потенциал для дальнейшего расширения.

Эти аспекты обеспечивают востребованность продукта среди целевой аудитории и создают основу для его успешного продвижения на рынке.

Однако, проект также имеет слабые стороны, такие как зависимость от актуальности предоставляемых данных, что требует регулярного обновления контента, а также высокая конкуренция среди аналогичных приложений.

Эти факторы могут создавать дополнительные вызовы, которые необходимо учитывать при разработке и продвижении продукта.

Для успешного развития приложения важно максимально эффективно использовать выявленные возможности.

К таким возможностям относятся установление партнёрств с образовательными и культурными организациями, расширение функционала приложения (например, добавление новых языков или уровней сложности [12]) и создание локализованных версий для разных регионов.

При этом необходимо минимизировать риски, связанные с потенциальными угрозами, такими как изменения в законодательстве, касающиеся работы приложений, или появление новых сильных конкурентов.

Для этого следует обеспечить гибкость и адаптивность архитектуры приложения, а также своевременно отслеживать изменения внешней среды.

Стратегия развития должна быть направлена на постоянное улучшение качества приложения [20], регулярное обновление контента и функционала, а также на расширение географического охвата.

Важной составляющей является активное продвижение продукта с учётом конкурентной среды, включая использование современных маркетинговых инструментов, таких как социальные сети, партнёрские программы и обратная связь от пользователей.

Особое внимание следует уделить обеспечению стабильности и безопасности приложения, а также поддержке пользователей.

Высокий уровень клиентского сервиса, оперативное решение возникающих проблем и учёт пожеланий пользователей создадут положительный имидж и укрепят позиции приложения на рынке..

VCM-анализ[11] приложения «Korean Simple» помогает оценить его конкурентоспособность, сопоставляя создаваемую ценность, затраты на разработку и размер рынка.

В силу того, что некоторые показатели сложно оценить количественно на этапе проектирования, анализ будет носить в основном качественный характер.

В таблице 5 представлен VCM-анализ.

Таблица 5 - VCM-анализ

	VCM-анализ разрабатываемой системы
V	<ol style="list-style-type: none"> 1. Экономическая ценность. 2. Высокая информационная ценность. 3. Удобство и простота использования.
C	<ol style="list-style-type: none"> 1. Маркетинг и продвижение. 2. Разработка и поддержка системы.
M	<ol style="list-style-type: none"> 1. Сегментация рынка. 2. Размер рынка. 3. Целевая аудитория.

VCM-анализ показал, что приложение «Korean Simple» имеет потенциал для успеха на рынке. Высокая ценность и значительный размер целевого рынка компенсируют затраты на разработку и поддержку приложения. Для достижения успеха необходимо грамотное управление затратами, а также проведение маркетинговой кампании и постоянное совершенствование приложения.

2 Проектирование приложения

2.1 Язык моделирования UML

В процессе проектирования приложения «Korean Simple» используется язык моделирования UML [5] для визуализации и спецификации различных аспектов системы. UML позволяет создавать наглядные модели, облегчающие понимание приложения как абстрактной сущности:

- диаграмма классов;
- диаграмма компонентов;
- диаграмма развёртывания;
- диаграмма вариантов использования;
- диаграммы последовательности.

2.2 Диаграмма вариантов использования (USE-CASE)

Для приложения «Korean Simple» диаграмма вариантов использования изображена на рисунке 4.



Рисунок 4 - Диаграмма вариантов использования (USE-CASE)

Диаграмма вариантов использования [6] в UML показывает функциональность системы с точки зрения пользователей. Диаграмма показывает, как пользователи взаимодействуют с системой, используя различные варианты использования системы. Эта диаграмма помогает определить основные функции приложения и их взаимосвязи, а также служит основой для дальнейшего проектирования.

2.3 Сценарии на основе USE-CASE диаграммы

Примеры использования описывают шаг за шагом, как пользователи взаимодействуют с приложением. На начальном этапе проектирования лучше избегать сложных диаграмм и псевдокода, а вместо этого использовать понятные и простые текстовые форматы для описания действий пользователя и реакций системы.

Для организации сценариев лучше всего использовать таблицы, чтобы записать все возможные варианты взаимодействия.

Ниже, в таблицах 6-11, представлены сценарии для двух вариантов использования приложения «Korean Simple» – тест и словарь:

Таблица 6 - «Главный раздел сценария» выполнения варианта использования «Тест»

Действие актеров	Актер
Актеры	Пользователь
Цель	Проверить и обобщить знания корейского языка
Краткое описание	Пользователь отвечает на вопросы и получает количество правильных ответов.
Тип	Базовый
Ссылки на другие варианты использования	Вывод уведомления о результате

Таблица 7 - «Типичный ход событий» выполнения варианта использования «Тест»

Действие актеров	Актер
1. Запускает вкладку «Тест»	Пользователь
2. Отображает страницу с вопросами	Приложение
3. Выбирает один из ответов	Пользователь
4. После ответов на все вопросы выводит ответ	Приложение

Таблица 8 - «Раздел исключения» выполнения варианта использования «Поиск пунктов приема отходов»

Действие актеров	Отклик системы
1. Ответ не правильный	Сообщения нет. Ответ засчитывается неверным.
2. Пользователь прерывает тест до завершения.	Прогресс не сохраняется.

Таблица 9 - «Главный раздел сценария» выполнения варианта использования «Словарь»

Действие актеров	Актер
Актеры	Пользователь
Цель	Найти перевод слова на корейский язык
Краткое описание	Пользователь находит перевод слова на иностранный язык, выбрав его из списка.
Тип	Базовый

Таблица 10 - «Типичный ход событий» выполнения варианта использования «Прохождение теста»

Действие актеров	Актер
1. Запуск вкладки «Словарь»	Пользователь
2. Отображает список слов	Приложение
3. Получает информацию	Пользователь

Таблица 11 - «Раздел исключения» выполнения варианта использования «Прохождение теста»

Действие актеров	Отклик системы
1. Пользователь прерывает тест до завершения.	2. Прогресс не сохраняется.
3. Попытка повторного прохождения теста	

2.4 Диаграмма последовательности

Диаграмма последовательности UML - это диаграмма взаимодействия, которая показывает, как объекты взаимодействуют во времени. Диаграмма показывает, в каком порядке вызываются методы между объектами, какие объекты инициируют взаимодействие и какие объекты отвечают.

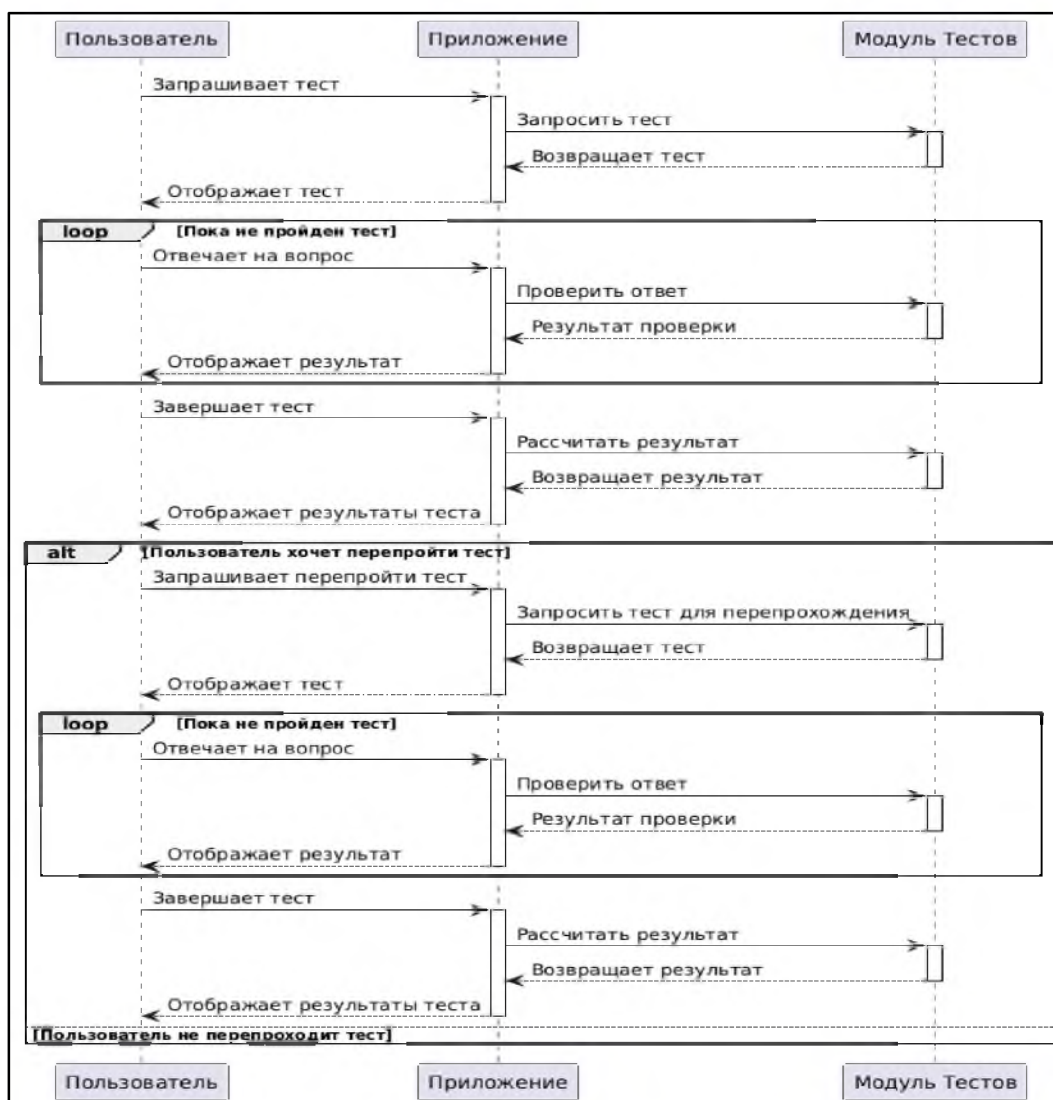


Рисунок 5. Диаграмма последовательности «Поиск приема отходов»

Диаграммы последовательностей [7] полезны для моделирования динамических аспектов системы. Диаграммы последовательностей часто используются для подробного описания сценариев использования.

Диаграмма на рисунке 5 описывает взаимодействие между объектами при поиске пользователем пунктов приема отходов.

Диаграмма на рис. 6 описывает взаимодействие между объектами при прохождении пользователем теста на знание правил утилизации отходов.

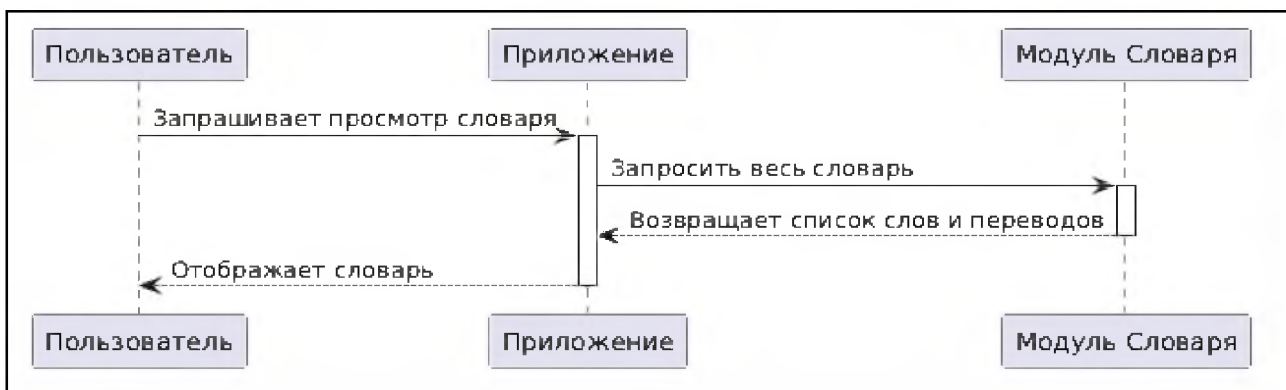


Рисунок 6 - Диаграмма последовательности «Прохождение теста»

2.5 Диаграмма классов

Диаграмма классов [8] в UML — это диаграмма, которая описывает структуру классов и их взаимодействие друг с другом (рисунок 7).

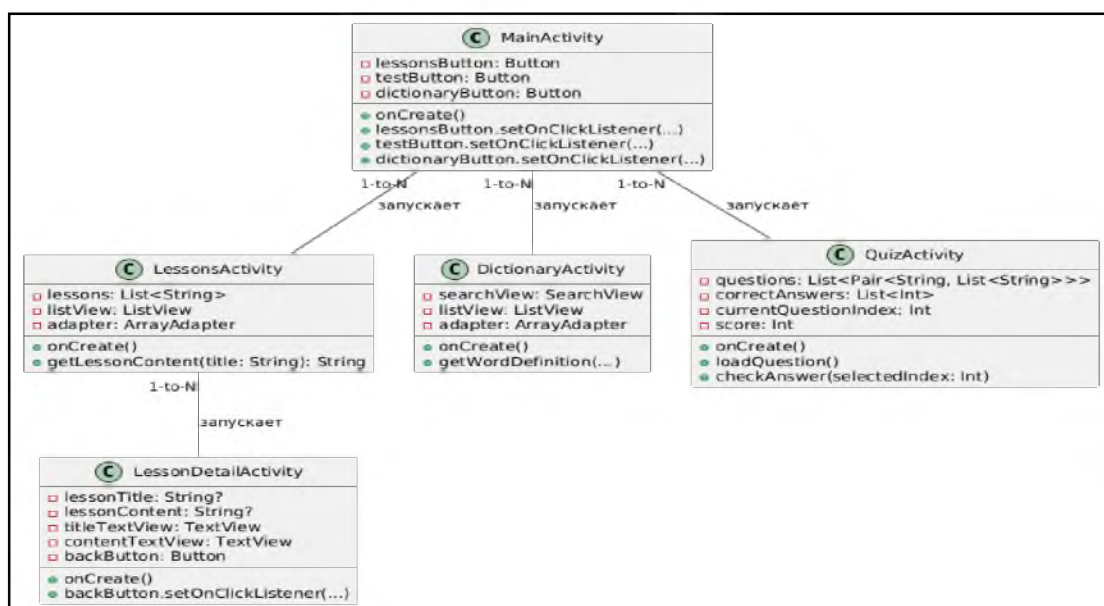


Рисунок 7 - Диаграмма классов

2.6 Диаграмма компонентов

Диаграммы компонентов и развертывания UML описывают архитектуру приложения на разных уровнях абстракции. В то время как диаграммы компонентов фокусируются на организации кода и зависимостях между модулями, диаграммы развертывания представляют физическое расположение компонентов в среде выполнения.

Диаграмма компонентов «Korean Simple» изображена на рисунке 8.

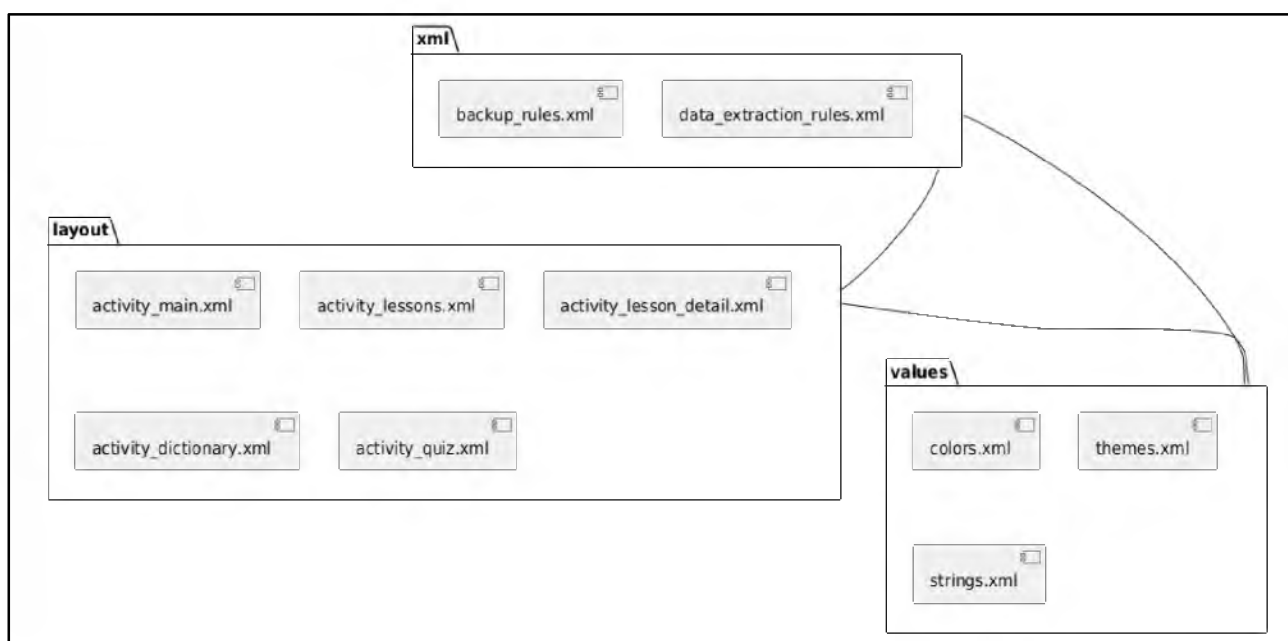


Рисунок 8 - Диаграмма компонентов

2.7 Диаграмма развертывания

Диаграммы компонентов отображают логическую структуру приложения, а диаграммы развертывания описывают его физическое размещение. В случае простого мобильного приложения диаграмма развертывания будет довольно простой, тогда как диаграмма компонентов наглядно демонстрирует более сложное взаимодействие внутренних модулей и функциональных частей приложения.

Диаграмма на рисунке 9 иллюстрирует, каким образом компоненты приложения распределены в его физической среде.

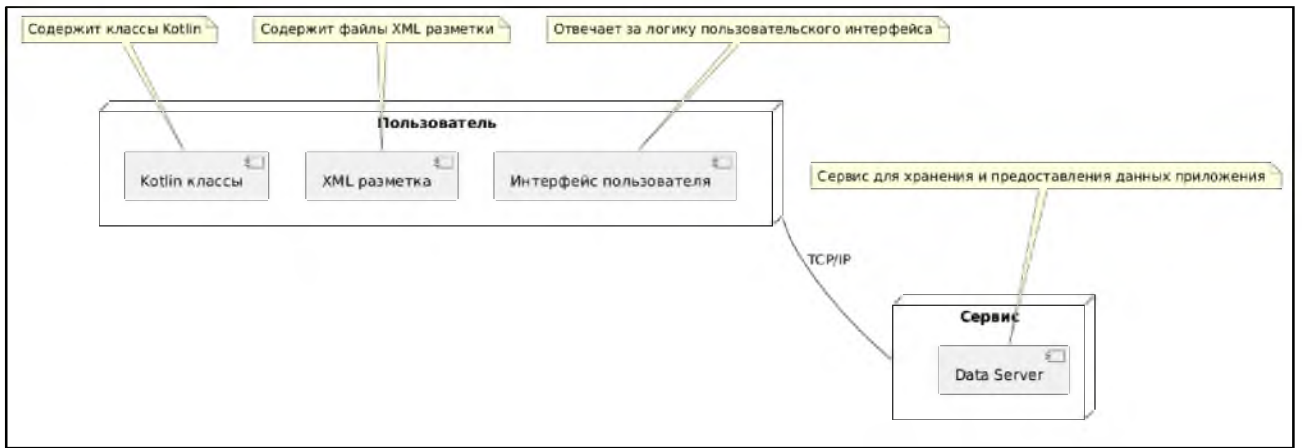


Рисунок 9 - Диаграмма развертывания

2.8 Модель архитектуры C4

Модель архитектуры C4 [17] представляет собой простой и действенный способ описания архитектуры программной системы, основанный на использовании четырех уровней абстракции. Каждый уровень отражает архитектуру с определенной точки зрения — от общего представления до детальных аспектов. Это делает модель понятной и полезной как для технических специалистов, так и для бизнес-заказчиков.

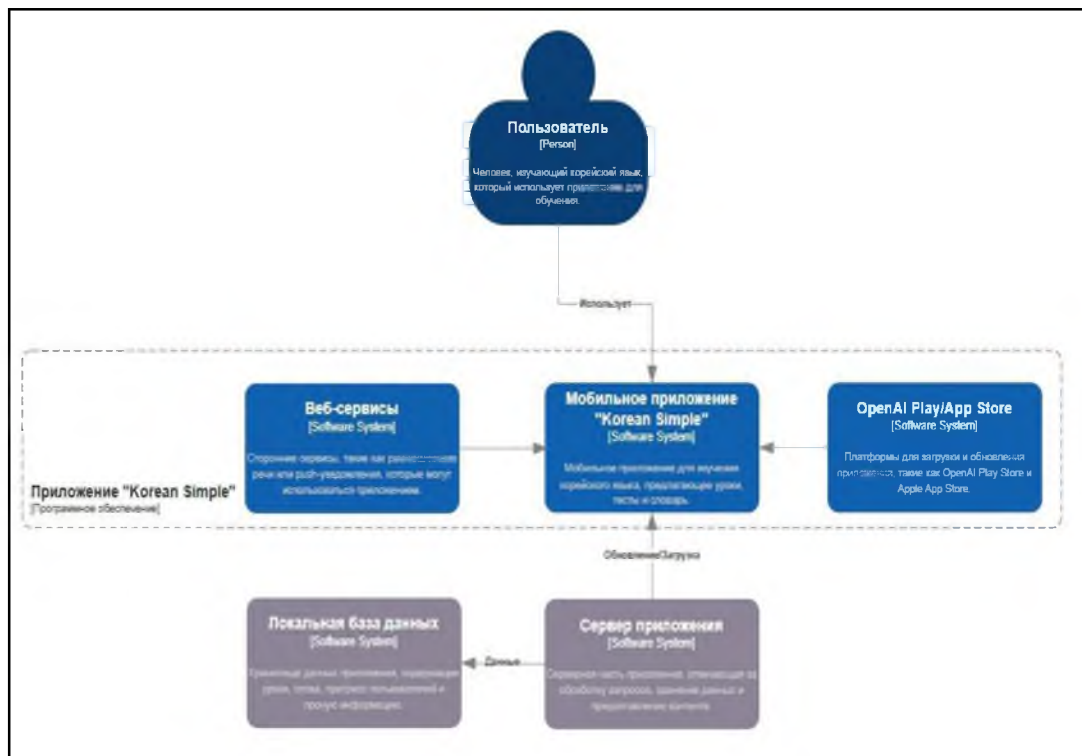


Рисунок 10. Диаграмма контекста

Наивысший уровень абстракции — контекст, который демонстрирует взаимодействие системы с внешними элементами (пользователями, другими системами) и отвечает на вопрос: «Что делает система и с чем она взаимодействует?». На рисунке 10 представлен пример такой диаграммы.

Диаграммы контейнеров описывают структуру системы, декомпозированная на контейнеры. Каждый контейнер содержит несколько компонентов как показано на рисунке 11.

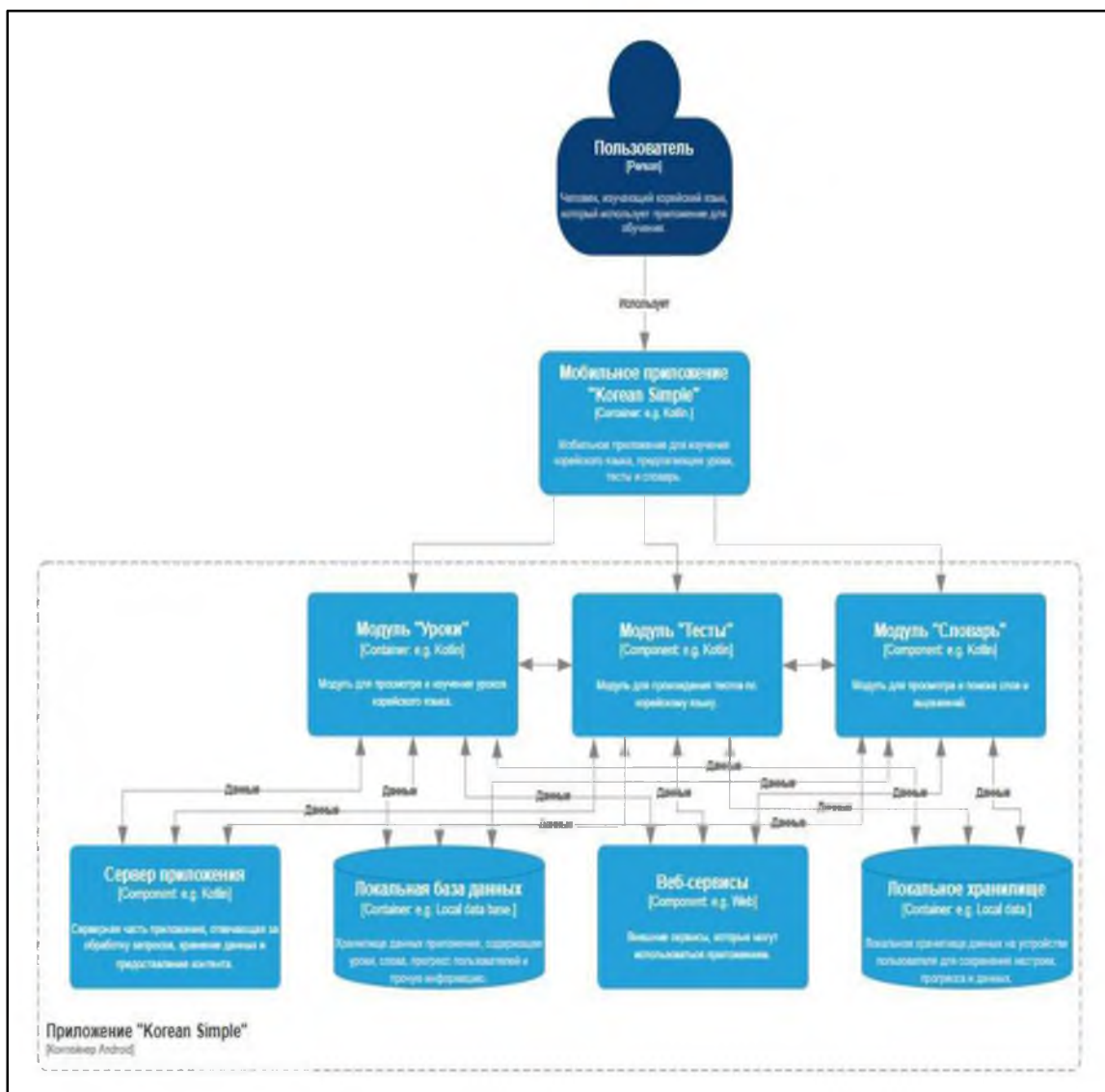


Рисунок 11 - Диаграмма контейнеров

Диаграмма компонентов описывает внутреннюю структуру контейнеров, показывая компоненты и их взаимодействия между собой. Это более детальный уровень абстракции. Диаграмма показано на рисунке 12.

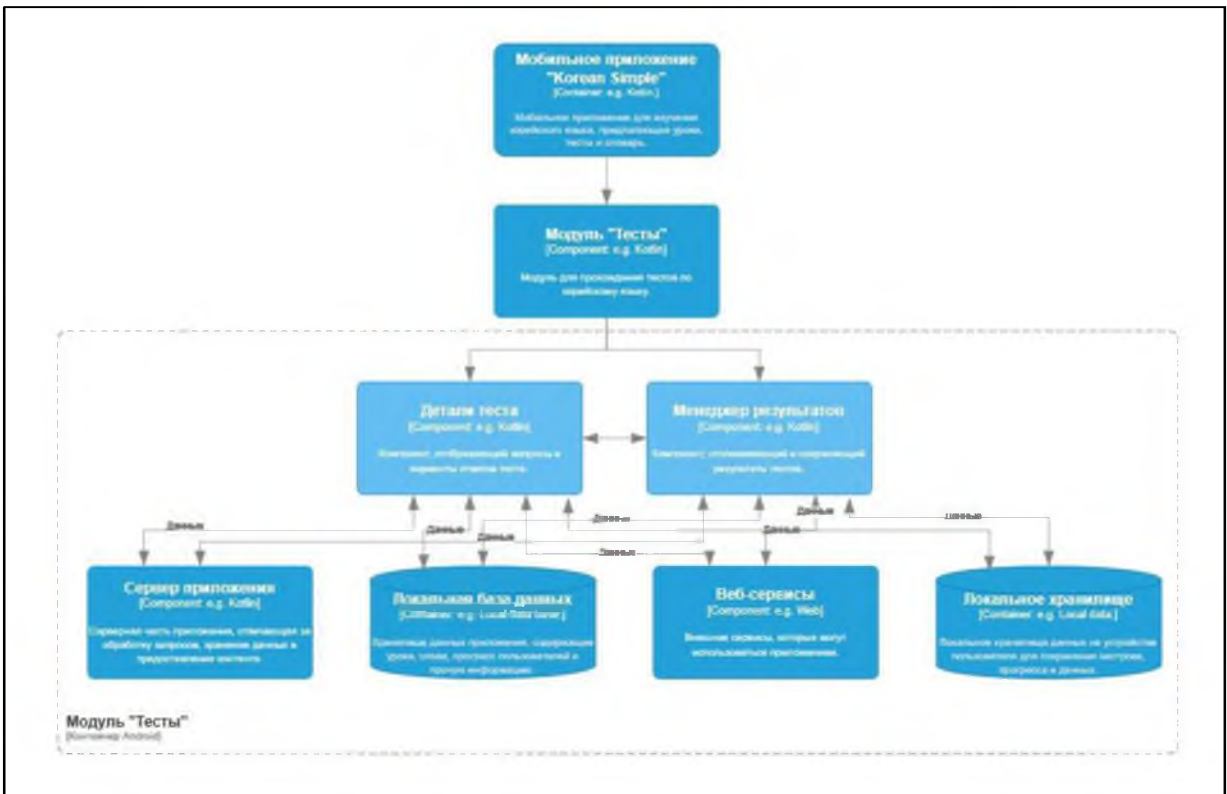


Рисунок 12 - Диаграмма компонентов

Диаграмма кода. Наименее абстрактная диаграмма. Она описывает детали реализации отдельных компонентов, таких как классы и функции, на уровне кода. Это показано на рисунке 13.

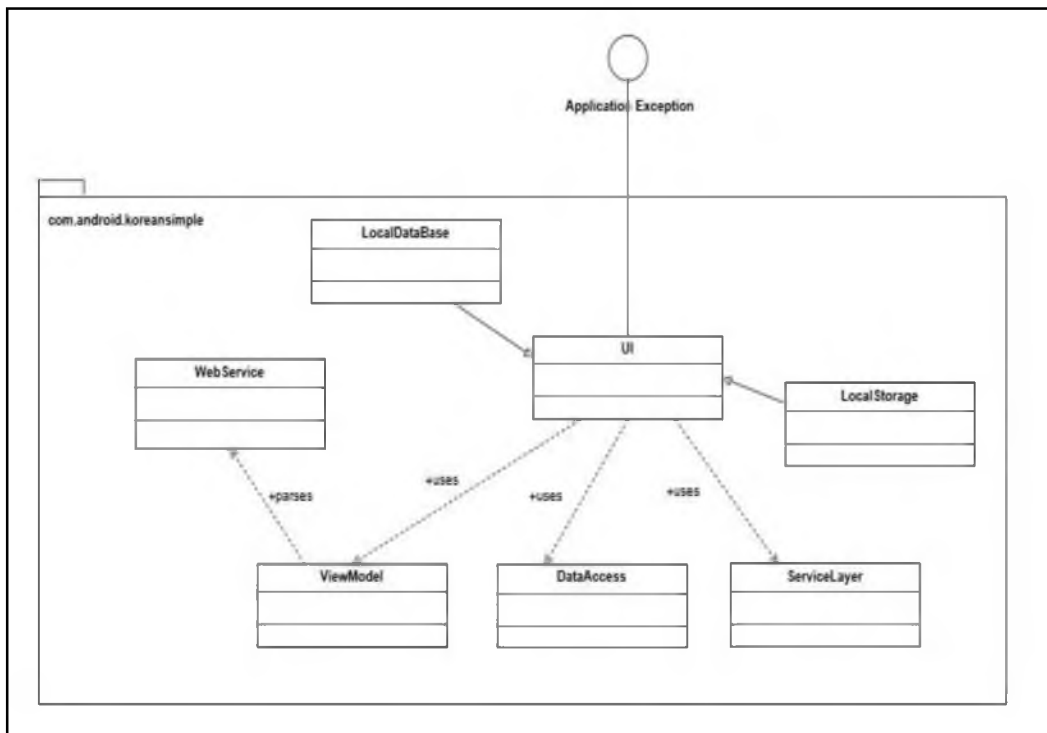


Рисунок 13 - Диаграмма кода

2.9 Декомпозиция приложения

Декомпозиция IDEF0 [16] представляет собой метод структурирования, который визуально отображает процесс в виде иерархии входов, выходов, управляющих данных и взаимосвязанных функций. Основное внимание уделяется описанию «что» делает процесс, а не «как» он выполняется. В отличие от блок-схем, демонстрирующих порядок выполнения операций, IDEF0 акцентирует внимание на функциональности. Этот метод использует иерархическую структуру, при которой процессы последовательно разбиваются на подпроцессы до уровня, обеспечивающего их полное понимание. Каждый уровень отражает определенную степень детализации.

Функциональные блоки в IDEF0 соединяются стрелками, указывающими направление потока данных. Декомпозиция системы обычно представляется в виде иерархического дерева, где верхний уровень соответствует системе в целом, а последующие уровни – выделенным подсистемам.

Первоначально IDEF0 был разработан для отображения процессов, процедур и деятельности внутри организации. Основным элементом нотации является графическое представление, обеспечивающее передачу ключевой информации. Этот подход способствует пониманию и анализу процессов, выявлению логики изменений, формированию требований к проекту и поддержке задач проектирования и интеграции на уровне системы.

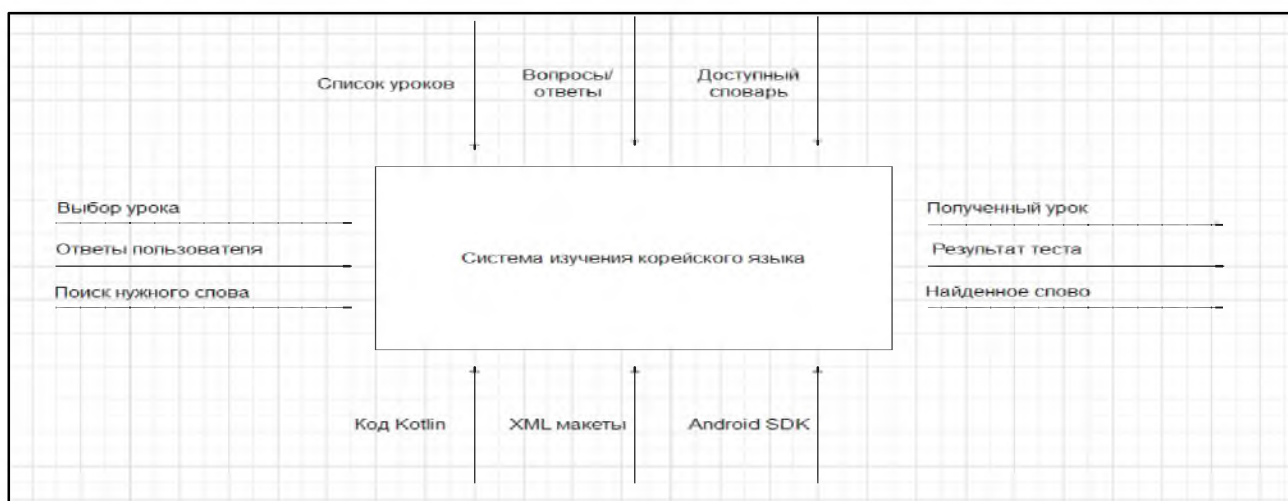


Рисунок 14 - Уровень 0 (A-0): Функция «Система утилизации отходов»

На рисунке 14 показан уровень 0 (A-0) с описанием функций системы управления отходами, а также декомпозиция уровня 1 (A-1), отражающая функции «системы управления отходами» на рисунке 15.

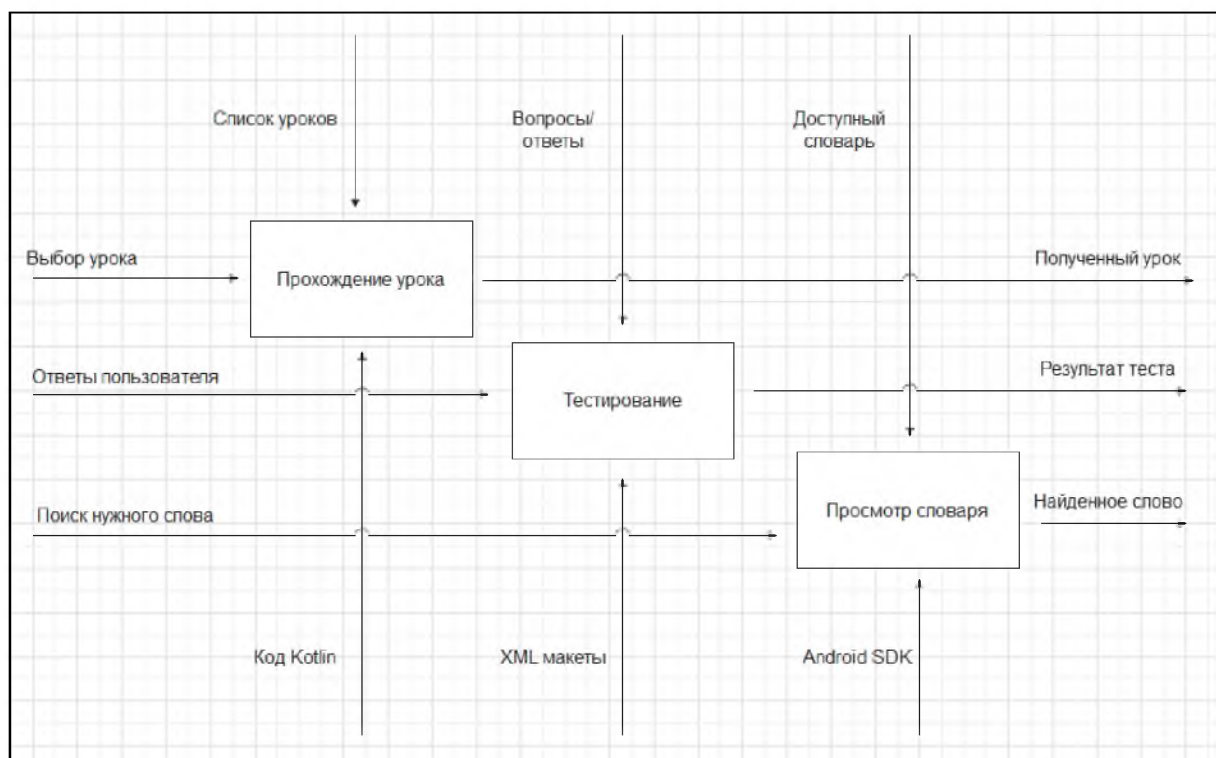


Рисунок 15 - Уровень 1 (A-1): Декомпозиция функции «Система утилизации ОТХОДОВ»

3 Разработка приложения

3.1 Описание выбранных средств разработки и языка программирования

Сильными сторонами проекта являются комплексный функционал приложения, актуальность выбранной темы и значительный потенциал для дальнейшего расширения. Эти аспекты обеспечивают востребованность продукта среди целевой аудитории и создают основу для его успешного продвижения на рынке.

Однако, проект также имеет слабые стороны, такие как зависимость от актуальности предоставляемых данных, что требует регулярного обновления контента, а также высокая конкуренция среди аналогичных приложений. Эти факторы могут создавать дополнительные вызовы, которые необходимо учитывать при разработке и продвижении продукта.

Для успешного развития приложения важно максимально эффективно использовать выявленные возможности. К таким возможностям относятся установление партнёрств с образовательными и культурными организациями, расширение функционала приложения (например, добавление новых языков или уровней сложности) и создание локализованных версий для разных регионов.

При этом необходимо минимизировать риски, связанные с потенциальными угрозами, такими как изменения в законодательстве, касающиеся работы приложений, или появление новых сильных конкурентов. Для этого следует обеспечить гибкость и адаптивность архитектуры приложения, а также своевременно отслеживать изменения внешней среды.

Стратегия развития должна быть направлена на постоянное улучшение качества приложения, регулярное обновление контента и функционала, а также на расширение географического охвата. Важной составляющей является активное продвижение продукта с учётом конкурентной среды, включая использование современных маркетинговых инструментов, таких как социальные сети, партнёрские программы и обратная связь от пользователей.

Особое внимание следует уделить обеспечению стабильности и безопасности приложения, а также поддержке пользователей. Высокий уровень клиентского сервиса, оперативное решение возникающих проблем и учёт пожеланий пользователей создадут положительный имидж и укрепят позиции приложения на рынке.

3.2 Описание основных алгоритмов приложения

Класс MainActivity - это основной активности (активности — это компонент Android, представляющий собой экран приложения) приложения «Korean Simple». Самая важная функция — onCreate(), которая инициализирует пользовательский интерфейс и обрабатывает нажатия на элементы нижней навигационной панели, как показано на рисунке 16.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val lessonsButton: Button = findViewById(R.id.button_lessons)
    val testButton: Button = findViewById(R.id.button_test)
    val dictionaryButton: Button = findViewById(R.id.button_dictionary)

    lessonsButton.setOnClickListener {
        startActivity(Intent(packageContext: this, LessonsActivity::class.java))
    }

    testButton.setOnClickListener {
        startActivity(Intent(packageContext: this, QuizActivity::class.java))
    }

    dictionaryButton.setOnClickListener {
        startActivity(Intent(packageContext: this, DictionaryActivity::class.java))
    }
}
```

Рисунок 16 - Алгоритм функции onCreate() класса MainActivity

Функция onCreate() устанавливает основной макет (activity_main), находит элементы button_lessons, button_text, button_dictionary, и регистрирует обработчик событий для её элементов. Когда пользователь нажимает на один из элементов навигации (уроки, тест, словарь), setOnClickListener запускает соответствующую активность (через startActivity()), таким образом осуществляется переключение экранов приложения, что является основным механизмом навигации между различными экранами приложения.

QuizActivity представляет собой экран, предназначенный для проведения викторины. Основная задача, реализуемая в методе onCreate(), заключается в отображении вопросов и обработке ответов пользователя с последующим переходом к следующему вопросу, как показано на рисунке 17.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_quiz)

    val questionText: TextView = findViewById(R.id.question_text)
    val answerButton1: Button = findViewById(R.id.answer_button_1)
    val answerButton2: Button = findViewById(R.id.answer_button_2)
    val answerButton3: Button = findViewById(R.id.answer_button_3)

    fun loadQuestion() {
        if (currentQuestionIndex < questions.size) {
            val question = questions[currentQuestionIndex]
            questionText.text = question.first
            answerButton1.text = question.second[0]
            answerButton2.text = question.second[1]
            answerButton3.text = question.second[2]
        } else {
            Toast.makeText(context: this, text: "Вы завершили тест! Ваш результат: $score/${questions.size}", Toast.LENGTH_LONG).show()
            finish()
        }
    }

    fun checkAnswer(selectedIndex: Int) {
        if (selectedIndex == correctAnswers[currentQuestionIndex]) {
            score++
        }
        currentQuestionIndex++
        loadQuestion()
    }

    answerButton1.setOnClickListener { checkAnswer(selectedIndex: 0) }
    answerButton2.setOnClickListener { checkAnswer(selectedIndex: 1) }
    answerButton3.setOnClickListener { checkAnswer(selectedIndex: 2) }

    loadQuestion()
}
```

Рисунок 17 - Алгоритм функции onCreate() класса QuizActivity

`onCreate()` в `QuizActivity` отвечает за инициализацию интерфейса викторины, включая отображение вопроса, вариантов ответов и кнопки «Далее». Кроме того, он подготавливает приложение к работе, загружая вопросы из списка `questions`. В полной версии реализация будет включать обработку пользовательских ответов и управление последовательным переходом между вопросами.

`DictionaryActivity`— это активити, отображающее словарь слов и фраз. Главная функция – `onCreate()`, которая создает экран с словарём и реализует возможность поиска по словам , как показано на рисунке 18.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_dictionary)

    val words = listOf(...)
    val listView: ListView = findViewById(R.id.dictionary_list)
    val searchView: SearchView = findViewById(R.id.search_view)
    val adapter = ArrayAdapter(context: this, android.R.layout.simple_list_item_1, words)
    listView.adapter = adapter

    searchView.setOnQueryTextListener(object : SearchView.OnQueryTextListener {
        override fun onQueryTextSubmit(query: String?): Boolean {
            return false
        }

        override fun onQueryTextChange(newText: String?): Boolean {
            adapter.filter.filter(newText)
            return false
        }
    })
}
```

Рисунок 18 - Алгоритм функции `onCreate()` класса `RecyclingPointsActivity`

`onCreate()` в `DictionaryActivity` отвечает за первоначальную настройку словаря. Она инициализирует пользовательский интерфейс, включая список

слов и строку поиска. При запуске активности список загружается из предварительно подготовленного массива или базы данных. Строка поиска позволяет пользователю быстро находить нужные слова или фразы по введенному тексту. При вводе запроса список автоматически фильтруется, показывая только совпадающие элементы.

Класс `LessonsActivity`— это активность, отображающее список уроков корейского языка. Его главная функция, `onCreate()`, загружает советы из источника данных (`getLessonContent()`) и выводит их на экран в `ListView`, как показано на рис. 19 стр. 38.

`onCreate()` в `LessonsActivity` инициализирует экран, отображающий список статей. Она получает ссылку на `ListView` из `layout` файла, а затем заполняет этот `ListView` текстом, содержащим сокращенную информацию об уроках. Таким образом, она подготавливает и выводит на экран информацию об уроках.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_lessons)

    val lessons = listOf("Урок 1: Знакомство с Хангылем (Корейским алфавитом) и базовыми гласными", "Урок 2: Базовые согласные и слоги", "Урок 3: Приветствия и ",
    , "Урок 4: Числительные (1-10)", "Урок 5: Частицы 은/는 и 이/가", "Урок 6: Глаголы и основы глаголов", "Урок 7: Настоящее время (вежливая форма) - 아/어요", "Урок 8

    val listView: ListView = findViewById(R.id.lessons_list)
    val adapter = ArrayAdapter<context: this, android.R.layout.simple_list_item_1, lessons>
    listView.adapter = adapter

    listView.setOnItemClickListener = AdapterView.OnItemClickListener { _, _, position, _ ->
        val intent = Intent<packageContext: this, LessonDetailActivity::class.java>
        intent.putExtra<name: "lesson_title", lessons[position]>
        intent.putExtra<name: "lesson_content", getLessonContent(lessons[position])>
        startActivity(intent)
    }
}
```

Рисунок 19 - Алгоритм функции `onCreate()` класса `LessonsActivity`

Класс LessonDetailActivity представляет собой активити, представляющее полную версию уроки.

onCreate() в LessonDetailActivity получает текст уроки и заголовок из LessonsActivity и отображает полную версию урока используя TextView. Вследствие этого на экране отображается полная версия урока и кнопка «назад» как показано на рис. 20.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_lesson_detail)

    val lessonTitle: String? = intent.getStringExtra( name: "lesson_title")
    val lessonContent: String? = intent.getStringExtra( name: "lesson_content")

    val titleTextView: TextView = findViewById(R.id.lesson_title)
    val contentTextView: TextView = findViewById(R.id.lesson_content)
    val backButton: Button = findViewById(R.id.button_back)
    titleTextView.text = lessonTitle
    contentTextView.text = lessonContent
    backButton.setOnClickListener {
        finish()
    }
}
```

Рисунок 20 - Алгоритм функции onCreate() класса LessonDetailActivity

3.3 Описание программных модулей

«Korean Simple» состоит из пяти активити. Все файлы расположены в директории kotlin+java, как показано на рисунке 21.

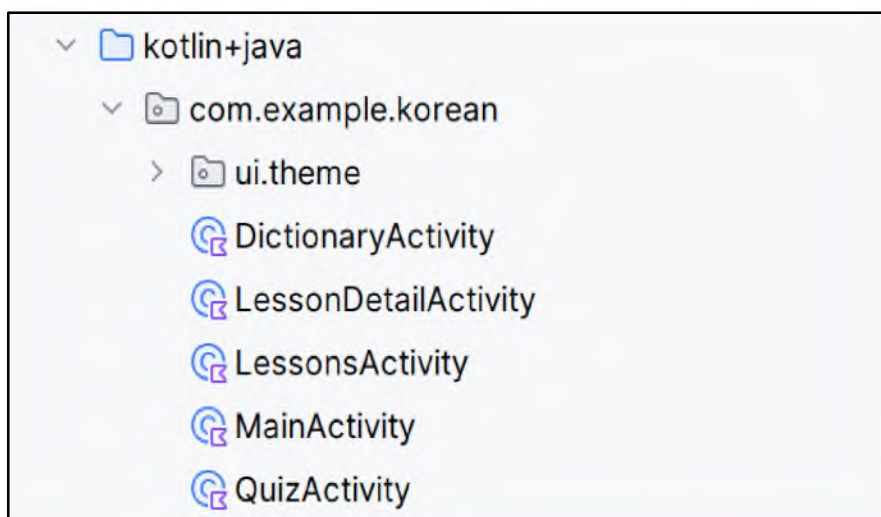


Рисунок 21 - Основные модули проекта

За визуальную составляющую приложения «Korean Simple» отвечают XML-файлы разметки, расположенные в ресурсах проекта. Структура изображена на рисунке 22. Эти файлы описывают визуальную составляющую каждого активити.

XML позволяет создавать чистый и структурированный стиль отображения, который легко понять и изменить. XML-файлы, используют стандартные Android виджеты и особенные компоненты для работы с приложением, обеспечивая ожидаемый пользовательский опыт.

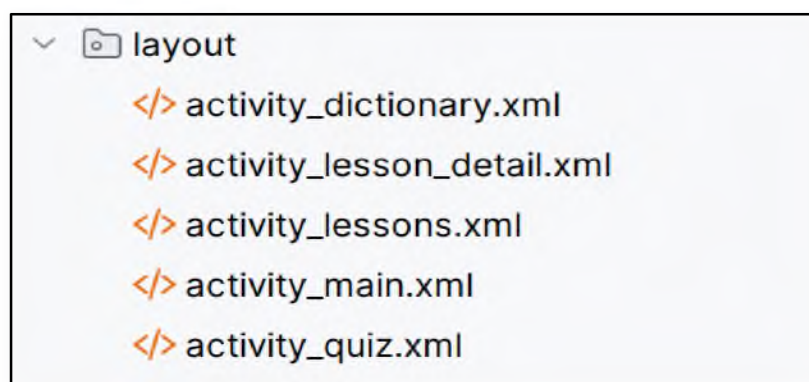


Рисунок 22 - Ресурсы проекта

Файл `activity_main.xml`, изображенный на рисунке 23-24 демонстрирует пример XML-разметки, определяющей положение и внешние особенности элементов интерфейса активити MainActivity.

В этом XML-файле описаны элементы для отображения меню с кнопками перехода на другие активити.

Анализ содержит подробное описание иерархии виджетов и их свойств, что позволяет лучше понять структуру данного конкретного экрана приложения.

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   android:layout_width="match_parent"
3   android:layout_height="match_parent"
4   android:orientation="vertical"
5   android:gravity="start"
6   android:padding="16dp"
7   android:background="@drawable/bg">
8
9   <TextView
10    android:layout_width="wrap_content"
11    android:layout_height="wrap_content"
12    android:text="Обучение корейскому языку"
13    android:textColor="@color/black"
14    android:fontFamily="sans-serif-black"
15    android:gravity="center"
16    android:textSize="38dp"
17    android:layout_marginBottom="26dp" />
18
19   <ImageView
20    android:layout_gravity="center"
21    android:layout_width="350dp"
22    android:layout_height="220dp"
23    android:background="@drawable/korea"></ImageView>
24
25   <Button
26    android:id="@+id/button_lessons"
27    android:layout_width="match_parent"
28    android:layout_height="78dp"
29    android:text="Уроки"
30    android:fontFamily="sans-serif-light"
31    android:layout_marginTop="26dp"
32    android:textSize="28dp"
33    />
34
35   <Button
36    android:id="@+id/button_test"
37    android:layout_width="match_parent"
38    android:layout_height="78dp"
39    android:text="Тесты"
40    android:fontFamily="sans-serif-light"
41    android:layout_marginTop="26dp"
42    android:textSize="28dp"/>
43
44   <Button
45    android:id="@+id/button_dictionary"
46    android:layout_width="match_parent"
47    android:layout_height="78dp"
48    android:text="Словарь"
49    android:fontFamily="sans-serif-light"
50    android:layout_marginTop="26dp"
51    android:textSize="28dp"/>
52 </LinearLayout>
```

Рисунок 23 - XML разметка код

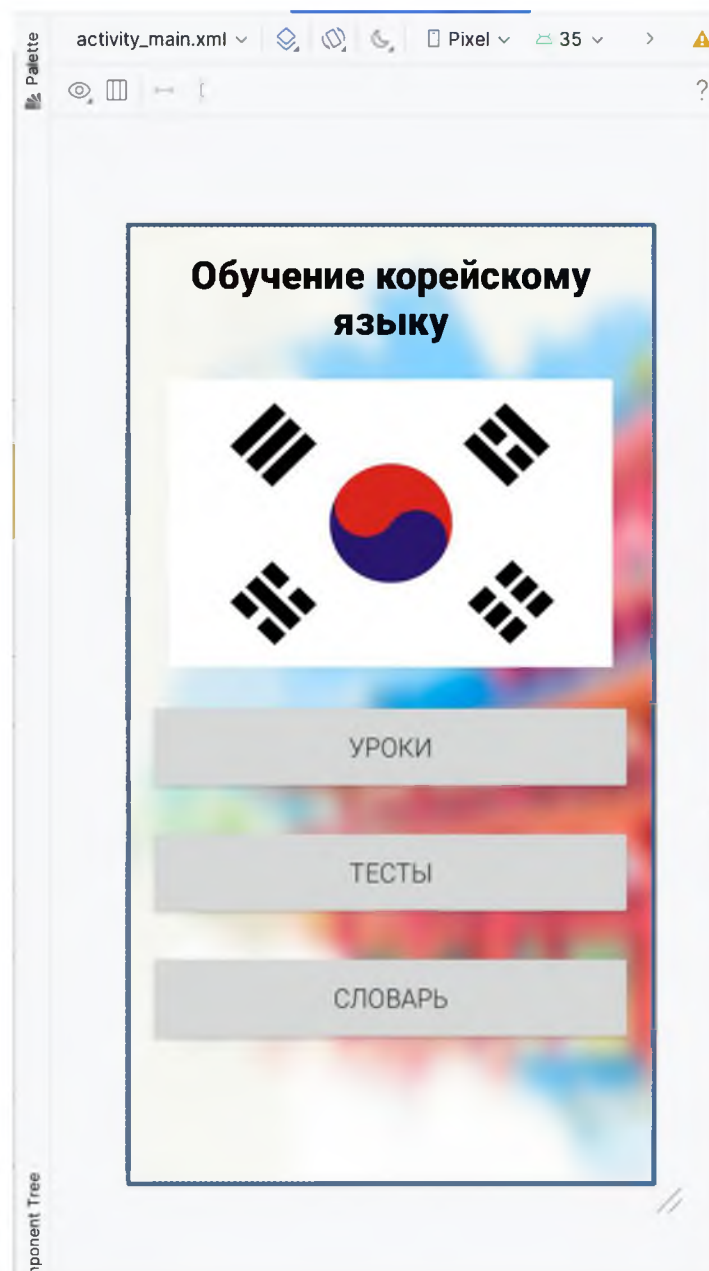


Рисунок 24 - XML разметка activity

3.4 Описание интерфейса приложения

Приложение «Korean Simple» предназначено для обучения корейскому языку. Интерфейс приложения состоит из нескольких экранов [9].

Фоновое изображение приложения неизменно на всех экранах и изображает корейский ханок.

Главный экран представляет собой список кнопок-переходов на другие экраны. В алгоритме используется `setOnClickListener` для перехода между

экранами. Элементы списка могут быть оформлены в виде нижнего горизонтального списка для улучшения понимания экрана..

Экран списка уроков представляет собой ListView с названиями и кратким описанием уроков.

Экран полной версии урока представляет собой полный текст урока с возможностью пролистывания контента и кнопку «Назад» для возврата на экран списка уроков.

Экран теста представляет собой вопрос и выбор вариантов ответа, а ниже присутствует кнопка сохранения ответа на вопрос. После окончания всех вопросов пользователь автоматически попадает на главный экран и в toast выводится количество правильных ответов.

Экран словаря представляет собой список слов и их переводы на русский язык. Предусмотрена возможность поиска слов путём ввода части слова в поле вверху экрана.

Интерфейс минималистичный и интуитивно понятный, ориентированный на простоту использования. Визуальное оформление должно быть чистым, с использованием ясной иконографии. Цветовая гамма спокойная и не отвлекающая внимание от основной информации. Навигация между экранами должна быть простой и очевидной. Для лучшего восприятия возможно использование иллюстраций или фотографий в дополнение к тексту.

Преимущества хорошего интерфейса в данном приложении:

1. Интуитивность и простота использования. Пользовательский интерфейс, разработанный с учётом принципов минимализма, обеспечивает лёгкое освоение приложения даже для тех, кто не имеет опыта использования мобильных образовательных платформ. Понятная навигация помогает пользователям быстро найти нужные функции — уроки, тесты или словарь.

2. Удобство изучения языка. Хорошо продуманный интерфейс способствует удобству восприятия информации. Уроки структурированы так, чтобы пользователь мог сосредоточиться на изучении конкретного элемента

языка (алфавита, слов или грамматических конструкций), не отвлекаясь на сложные элементы дизайна.

3. Повышение мотивации. Интерфейс включает в себя элементы геймификации — такие как прогресс-бар, накопление очков и уровни, что стимулирует пользователей к регулярным занятиям и повышает их вовлечённость в процесс обучения.

4. Доступность для разной аудитории. Интерфейс разработан с учётом универсальных требований к дизайну, что делает приложение доступным для широкой аудитории — от школьников до взрослых пользователей. Лёгкость восприятия интерфейса способствует тому, что приложение подходит для использования в любом возрасте.

5. Скорость взаимодействия. Хорошо продуманный интерфейс обеспечивает высокую скорость выполнения задач. Пользователь может быстро переходить между уроками, тестами и словарём, не испытывая сложностей с загрузкой или поиском информации. Это экономит время и делает обучение более эффективным.

6. Эстетическое удовольствие. Приятный визуальный стиль, лаконичный дизайн и грамотное использование цветовой палитры создают комфортную среду для обучения. Эстетика интерфейса помогает удерживать внимание пользователя, снижая утомляемость и повышая удовлетворённость от работы с приложением.

7. Адаптивность и масштабируемость. Интерфейс разработан с учётом возможности дальнейшего масштабирования и добавления новых функций без потери удобства. Это важно для будущих версий приложения, которые могут включать новые курсы, уровни сложности и дополнительные модули.

Главный экран приложения содержит изображение корейского флага и ссылками-переходами на другие экраны. Шрифт должен быть понятным и большого размера, чтобы дать пользователю понятный интерфейс, что изображено на рисунке 25.

Пункты меню содержат ссылки для перехода на другие экраны, такие как:

Уроки: переход на экран с списком уроков.

Тест: начало прохождения квиза на 10 вопросов на тему корейского языка.

Словарь: переход на экран с словарём слов на корейском языке.

Визуально меню должно быть оформлено в том же стиле, что и все остальное приложение. Кнопки меню должны быть достаточно крупными и хорошо различимыми, чтобы обеспечить удобство использования на различных устройствах.

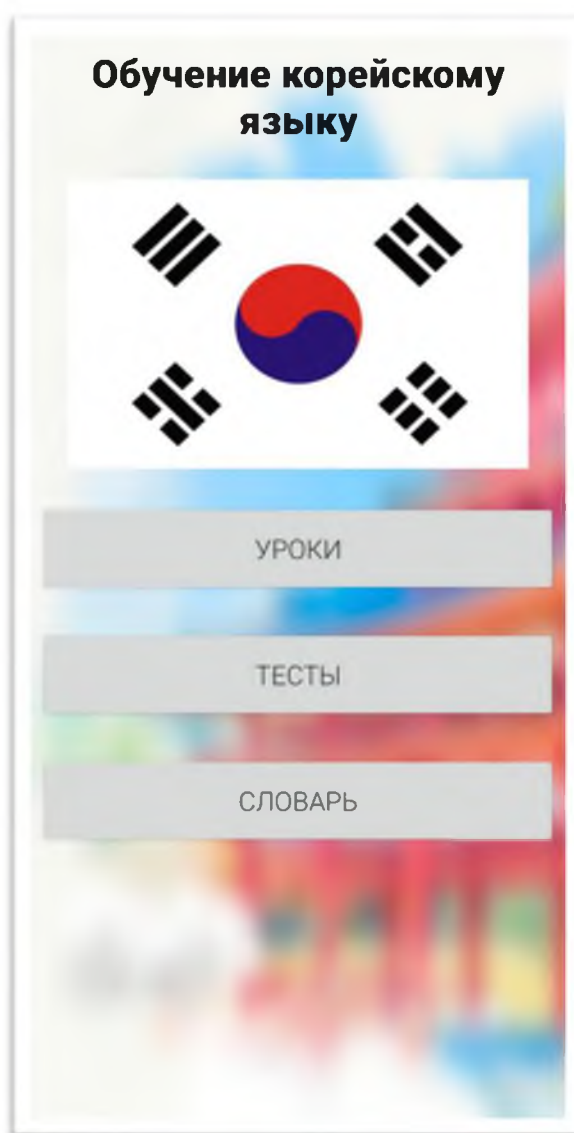


Рисунок 25 - Главное окно приложения

Экран «Уроки» отображает краткий список уроков, как показано на рис. 26. Каждый элемент списка содержит название урока контрастирует с

остальным оформлением проекта.

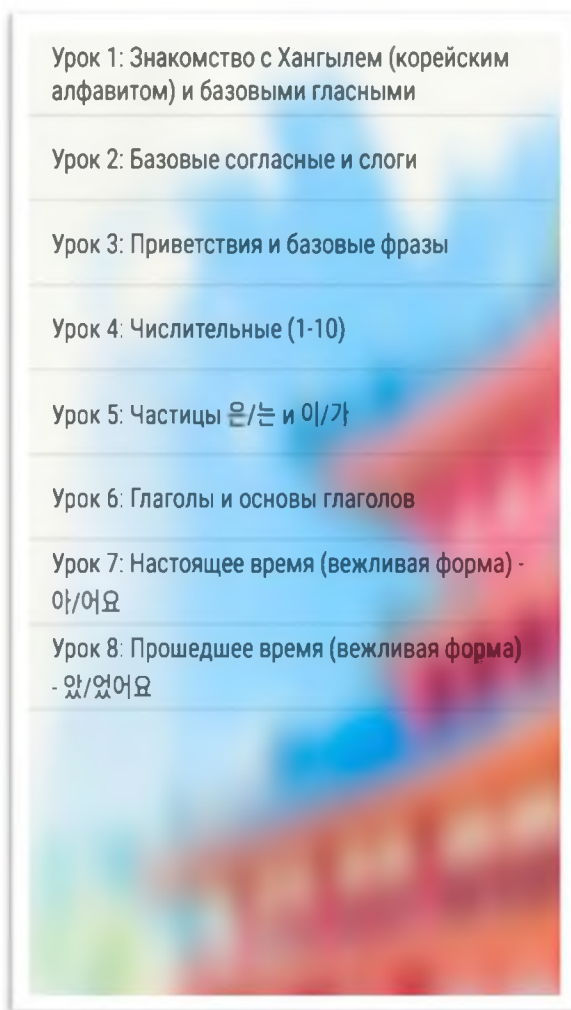


Рисунок 26 - Экран «Уроки»

Экран «Тест» на рисунке 27 стр. 46 предназначен для прохождения теста на знание корейского языка.

При выборе варианта ответа система проверяет его на корректность и сохраняет результат для последующего анализа статистики. Пользователь может увидеть, правильно ли он ответил, благодаря отображаемому toast-сообщению с кратким результатом.

После завершения всех вопросов теста система показывает итоговую оценку, общее количество правильных и неправильных ответов.

Также планируется добавить возможность отслеживать прогресс пользователя, включая сохранение результатов предыдущих тестов и отображение графика улучшения навыков.

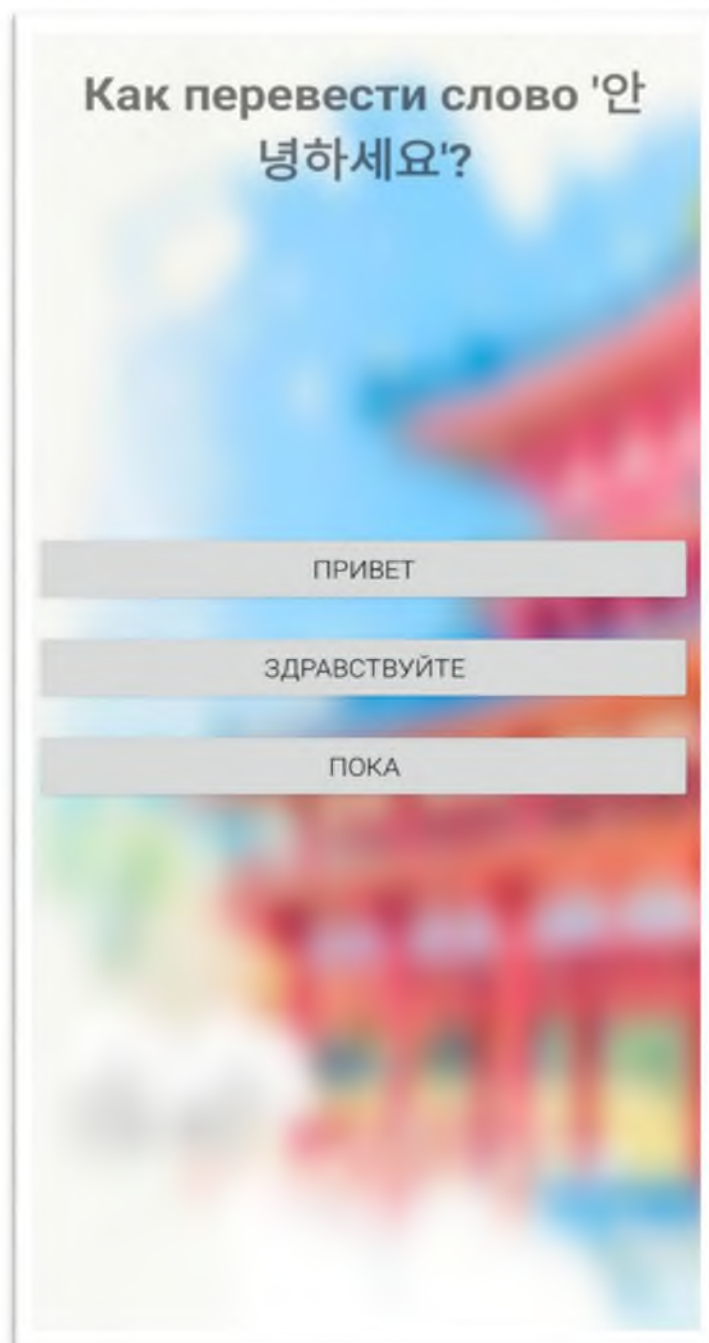


Рисунок 27 - Экран «Тест»

Экран «Словарь» предоставляет справочник с переводами популярных слов, как показано на рисунке 28.

Экран позволяет пользователю быстро находить нужные слова, используя строку поиска в верхней части интерфейса. Список слов подобран из списка популярных обиходных слов в Корейской речи [15].

В будущем планируется добавить возможность прослушивания произношения слов носителем языка, а также интеграцию с экраном «Тесты», где пользователь сможет проверить свои знания по изученным словам.



Рисунок 28 - Экран «Словарь»

Визуально экраны должны быть оформлены в том же стиле, что и другие экраны в приложении, с использованием четких и понятных шрифтов, достаточного свободного пространства и привлекательной цветовой гаммы. В целом, дизайн должен быть направлен на легкое распознавание и запоминание информации.

В заключение, интерфейс приложения «Korean Simple» разработан с учетом принципов удобства использования, доступности и визуальной

привлекательности. Все экраны последовательны по стилю и навигации, обеспечивая бесшовный пользовательский опыт. Ключевые элементы интерфейса легкодоступны и понятны для всех пользователей, независимо от уровня их технической подготовки.

Успешный интерфейс приложения «Korean Simple» будет способствовать облегчению освоения иностранного языка. Вся информация должна быть легко доступна и понятна, что значительно повышает эффективность приложения.

Заключение

Процесс разработки мобильного приложения для изучения корейского языка «Korean Simple» представляет собой ответ на растущий интерес к корейской культуре, языку и образовательным технологиям. Проект направлен на создание удобного и доступного инструмента для начинающих, который поможет пользователям освоить базовые элементы языка, включая алфавит, лексику и грамматику, а также закрепить знания с помощью практических упражнений.

В ходе выполнения проекта были достигнуты ключевые результаты, включающие разработку структуры приложения, реализацию интуитивно понятного пользовательского интерфейса и создание функционала, обеспечивающего эффективное взаимодействие пользователя с обучающим контентом. Приложение спроектировано с использованием современных технологий, включая язык программирования Kotlin и платформу Android, что обеспечивает его стабильность, производительность и возможность дальнейшего масштабирования.

Основное преимущество «Korean Simple» заключается в его ориентации на пользователя: минималистичный дизайн, адаптивный контент и геймифицированные элементы мотивируют к регулярным занятиям, делая обучение не только эффективным, но и увлекательным. В приложении реализована система отслеживания прогресса, что позволяет пользователям видеть свои достижения и усиливает чувство удовлетворения от обучения.

Проект также учитывает ключевые вызовы и риски, связанные с разработкой образовательных мобильных приложений, включая необходимость поддержания актуальности данных, конкуренцию на рынке и соответствие изменениям в законодательстве.

В рамках стратегии развития заложена гибкость архитектуры, что обеспечит возможность оперативного обновления контента и добавления новых функций.

Реализация «Korean Simple» открывает широкие перспективы. Возможности для партнёрства с образовательными учреждениями, добавление новых языковых курсов и адаптация приложения под разные культурные контексты создают условия для его дальнейшего развития. Расширение географического охвата и использование современных маркетинговых инструментов помогут привлечь новую аудиторию и укрепить позиции приложения на рынке.

В заключение, проект «Korean Simple» имеет значительный потенциал для популяризации изучения корейского языка, благодаря своему удобству, актуальности и инновационному подходу. Приложение станет полезным инструментом для всех, кто интересуется изучением корейского языка и культуры, и внесёт вклад в развитие образовательных технологий.

В приложении реализованы следующие ключевые модули:

1. Уроки корейского языка. Интерактивные занятия, которые помогают пользователям изучить основы языка, включая хангыль (корейский алфавит), базовую лексику и грамматику. Уроки построены с учётом последовательного освоения материала, что обеспечивает плавное и системное погружение в язык.

2. Тест на знание корейского языка. Функционал, позволяющий пользователям проверить свои знания и оценить прогресс. Тесты помогают определить уровень владения языком, выявить пробелы в знаниях и сосредоточиться на их устранении.

3. Словарь. Удобный инструмент, содержащий базовый набор слов и выражений с переводами и произношением. Словарь доступен в любое время и служит не только справочным ресурсом, но и средством для расширения словарного запаса.

Эти модули создают основу для комплексного подхода к изучению языка, объединяя теоретическое обучение, практическую проверку знаний и постоянное пополнение лексики.

Проект «Korean Simple» сочетает в себе актуальность и технологические решения, предоставляя пользователям простой и эффективный инструмент для

изучения корейского языка. В будущем приложение может быть дополнено новыми функциями, такими как расширенные тесты, аудиотренировки и тематические уроки. Это обеспечит ещё более глубокое погружение в язык и усилит образовательный эффект [13].

Таким образом, приложение не только решает актуальные задачи обучения корейскому языку, но и создаёт платформу для дальнейшего развития, предоставляя пользователям удобный и качественный инструмент для изучения языка и культуры Кореи.

В результате разработано функциональное и удобное приложение, способное существенно упростить процесс ознакомления с правилами утилизации отходов. «Korean Simple» предоставляет пользователю все необходимые инструменты для изучения корейского языка.

Все поставленные задачи выполнены в полном объеме, и проект успешно завершен.

Список литературы

1. 23+ лучших приложений для изучения английского языка 2024. [Электронный ресурс] — Режим доступа: <https://videoinfographica.com/learn-english-apps/#Simpler> (дата обращения 02.01.2025).
2. 23+ лучших приложений для изучения английского языка 2024. [Электронный ресурс] — Режим доступа: <https://videoinfographica.com/learn-english-apps/#PuzzleEnglish> (дата обращения 02.01.2025).
3. 23+ лучших приложений для изучения английского языка 2024. [Электронный ресурс] — Режим доступа: <https://videoinfographica.com/learn-english-apps/#LingoDeer>.(дата обращения 02.01.2025).
4. Как построить диаграмму Ганта в Excel: инструкция со скриншотами. [Электронный ресурс] — Режим доступа: <https://skillbox.com/media/management/kak-postroit-diagrammu-ganta-v-excel-instruktsiya-so-skrinshotami/>(дата обращения 03.01.2025).
5. Зачем нам UML? Или как сохранить себе нервы и время. [Электронный ресурс] — Режим доступа: <https://habr.com/ru/articles/458680/> (дата обращения 04.01.2025).
6. Диаграмма прецедентов (use case диаграмма). [Электронный ресурс] — Режим доступа: <https://testengineer.ru/use-case-diagram/> (дата обращения 04.01.2025).
7. Диаграмма последовательности (sequence-диаграмма). [Электронный ресурс] — Режим доступа: <https://habr.com/ru/articles/814769/> (дата обращения 03.01.2025).
8. Использование диаграммы классов UML при проектировании и документировании программного обеспечения. [Электронный ресурс] — Режим доступа: <https://habr.com/ru/articles/572234/>(дата обращения 03.01.2025).
9. Что такое UX/UI дизайн на самом деле? [Электронный ресурс] — Режим доступа: <https://habr.com/ru/articles/321312/>(дата обращения 04.01.2025).
10. На каком языке рисуют схемы: что такое UML и почему его понимают

во всём мире [Электронный ресурс] — Режим доступа: <https://practicum.yandex.ru/blog/uml-diagrammy/>(дата обращения 06.01.2025).

11. Value Conflict Mapping (VCM): Идентификация и решение системных конфликтов. [Электронный ресурс] — Режим доступа: <https://alekseichernysh.ru/slovar/tpost/4u1ugzt771-value-conflict-mapping-vcm-identifikatsi>(дата обращения 05.01.2025).

12. Шкала уровней сложности. [Электронный ресурс] — Режим доступа: <https://www.фонд-трионикс.рф/obrazovanie/shkala-urovney-slozhnosti-kursov/> (дата обращения 06.01.2025).

13. Образовательный эффект: сущность и механизм проявления в высшем образовании. [Электронный ресурс] — Режим доступа: <https://cyberleninka.ru/article/n/obrazovatelnyy-effekt-suschnost-i-mehanizm-proyavleniya-v-vysshem-obrazovanii> (дата обращения 25.11.2024).

14. К-поп. [Электронный ресурс] — Режим доступа: <https://ru.wikipedia.org/wiki/К-поп> (дата обращения 04.01.2025).

15. 250 слов на корейском языке, которые должен знать каждый. [Электронный ресурс] — Режим доступа: <https://www.divelang.ru/blog/useful/250-slov-na-korejskom-yazyke-kotorye-dolzhen-znat-kazhdyj/> (дата обращения 07.01.2025).

16. Азы моделирования в idef0. [Электронный ресурс] — Режим доступа: <https://www.comindware.ru/blog/%D0%B0%D0%B7%D1%8B-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%B2-idef0/>(дата обращения 07.01.2025).

17. Как описать большую систему в нотации С4. [Электронный ресурс] — Режим доступа: <https://habr.com/ru/companies/nspk/articles/679426/>(дата обращения 08.01.2025).

18. Курс по изучению Kotlin и Android Studio с нуля до Junior. #1 Введение. Установка Android Studio [Электронный ресурс] — Режим доступа: <https://habr.com/ru/articles/717074/>(дата обращения 06.01.2025).

19. Немного о Kotlin [Электронный ресурс] — Режим доступа:

<https://habr.com/ru/articles/277479/> (дата обращения 07.01.2025).

20. Как задавать требования к качеству ПО в цифрах?.[Электронный ресурс] — Режим доступа: <https://habr.com/ru/articles/661331/>(дата обращения 24.11.2024).

21. Мировая экономика в 2 ч. Часть 1. : учеб. для вузов / Р. И. Хасбулатов [и др.]; под редакцией Р. И. Хасбулатова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 689 с. — (Высшее образование). — ISBN 978-5-534-11204-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/542412> (дата обращения: 09.01.2025).

22. Соколова, В. В. Вычислительная техника и информационные технологии. Разработка мобильных приложений: учеб. пособие для вузов / В. В. Соколова. — Москва: Издательство Юрайт, 2024. — 160 с. — (Высшее образование). — ISBN 978-5-534-16302-5. — Текст: электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/537272> (дата обращения: 09.01.2025).

23. Соколова, В. В. Разработка мобильных приложений: учеб. пособие для среднего профессионального образования / В. В. Соколова. — Москва: Издательство Юрайт, 2024. — 160 с. — (Профессиональное образование). — ISBN 978-5-534-16868-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/542342> (дата обращения: 09.01.2025).

24. Сысолетин, Е. Г. Разработка интернет-приложений: учеб. пособие Е. Г. Сысолетин, С. Д. Ростунцев. — Москва: Издательство Юрайт, 2024. — 80 с. — ISBN 978-5-534-19603-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/556745> (дата обращения: 09.01.2025).

25. Тузовский, А. Ф. Объектно-ориентированное программирование : учеб. пособие для вузов / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2024. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст:

электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/537332> (дата обращения: 09.01.2025).

26. Прошина, З. Г. Межкультурная коммуникация: английский язык и культура народов Восточной Азии : учеб. пособие для вузов / З. Г. Прошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2024. — 399 с. — (Высшее образование). — ISBN 978-5-534-12403-3. — Текст: электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/541831> (дата обращения: 09.01.2025).

27. Методика обучения иностранному языку: учеб. и практикум для вузов / О. И. Трубицина [и др.] ; под редакцией О. И. Трубициной. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 457 с. — (Высшее образование). — ISBN 978-5-534-19070-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/555883> (дата обращения: 09.01.2025).

28. Протасова, Е. Ю. Методика раннего обучения иностранному языку : учебное пособие для вузов / Е. Ю. Протасова, Н. М. Родина; под редакцией Е. Ю. Протасовой. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2024. — 255 с. — (Высшее образование). — ISBN 978-5-534-11276-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/542354> (дата обращения: 09.01.2025).

29. Титова, С. В. Цифровая методика обучения иностранным языкам : учебник для вузов / С. В. Титова. — Москва: Издательство Юрайт, 2024. — 248 с. — (Высшее образование). — ISBN 978-5-534-16848-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL: <https://urait.ru/bcode/531883> (дата обращения: 09.01.2025).

30. Протасова, Е. Ю. Теория и методика развития речи у детей. Обучение двуязычных дошкольников: учеб. пособие /Е. Ю. Протасова, Н. М. Родина; под редакцией Е. Ю. Протасовой. — 2-е изд. — Москва: Издательство Юрайт, 2024. — 208 с. — ISBN 978-5-534-11189-7. — Текст: электронный //

Образовательная платформа Юрайт [сайт]. [Электронный ресурс]— URL:
<https://urait.ru/bcode/542397> (дата обращения: 09.01.2025).

Приложение 1

Листинг приложения

```
// MainActivity.kt
package com.example.korean

import android.app.Activity
import android.content.Intent
import android.os.Bundle

import android.widget.Button

class MainActivity : Activity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val lessonsButton: Button =
            findViewById(R.id.button_lessons)
        val testButton: Button =
            findViewById(R.id.button_test)
        val dictionaryButton: Button =
            findViewById(R.id.button_dictionary)

        lessonsButton.setOnClickListener {
            startActivity(Intent(this,
                LessonsActivity::class.java))
        }

        testButton.setOnClickListener {
            startActivity(Intent(this,
                QuizActivity::class.java))
        }

        dictionaryButton.setOnClickListener {
            startActivity(Intent(this,
                DictionaryActivity::class.java))
        }
    }
}

// QuizActivity.kt
package com.example.korean

import android.app.Activity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.widget.Toast

class QuizActivity : Activity() {
    private val questions: List<Pair<String, List<String>>>
        get() = listOf(
            "Как перевести слово '안녕하세요?' to listOf("Привет",
                "Здравствуйте", "Пока"),
            "Какое это слово: '감사합니다?' to listOf("Спасибо",
                "Простите", "Здравствуйте"),
            "Какой из этих символов представляет гласный звук
            'а'?" to listOf("ㅏ", "ㅑ", "ㅓ"),
            "Какой из этих символов представляет согласный звук
            'м'?" to listOf("ㄴ", "ㄷ", "ㄹ"),
            "Как читается слог 'ㄱ?' to listOf("ga", "na",
                "da"),
            "Какой символ не является согласным?" to listOf("ㄱ",
                "ㄴ", "ㅇ"),
            "Как правильно написать слог 'yo'?" to listOf("ㅇ",
                "ㅓ", "ㅑ"),
            "Как сказать 'Здравствуйте' на корейском
            (формально)?" to listOf("안녕", "안녕하세요", "죄송합니다"),
            "Как будет 'три' на корейском?" to listOf("하", "물",
                "셋"),
            "Какой цифре соответствует '0'?" to listOf("6",
                "5", "4"),
        )

    private val correctAnswers =
        listOf(1,0,1,1,2,2,1,1,2,0)

    private var currentQuestionIndex = 0
    private var score = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_quiz)

        val questionText: TextView =
            findViewById(R.id.question_text)
        val answerButton1: Button =
            findViewById(R.id.answer_button_1)
        val answerButton2: Button =
            findViewById(R.id.answer_button_2)
        val answerButton3: Button =
            findViewById(R.id.answer_button_3)

        fun loadQuestion() {
            if (currentQuestionIndex <
                questions.size) {
                val question =
                    questions[currentQuestionIndex]
                questionText.text = question.first
                answerButton1.text =
                    question.second[0]
                answerButton2.text =
                    question.second[1]
                answerButton3.text =
                    question.second[2]
            } else {
                Toast.makeText(this, "Вы завершили
                тест! Ваш результат: $score/${questions.size}",
                    Toast.LENGTH_LONG).show()
                finish()
            }
        }

        fun checkAnswer(selectedIndex: Int) {
            if (selectedIndex ==
                correctAnswers[currentQuestionIndex]) {
                score++
                currentQuestionIndex++
                loadQuestion()
            }

            answerButton1.setOnClickListener {
                checkAnswer(0) }
            answerButton2.setOnClickListener {
                checkAnswer(1) }
            answerButton3.setOnClickListener {
                checkAnswer(2) }

            loadQuestion()
        }
    }
}

// DictionaryActivity.kt
package com.example.korean

import android.app.Activity
import android.os.Bundle
```

```

import android.widget.AdapterView
import android.widget.ListView
import android.widget.SearchView

class DictionaryActivity : Activity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_dictionary)

        val words = listOf(
            )
        val listView: ListView =
findViewById(R.id.dictionary_list)
        val searchView: SearchView =
findViewById(R.id.search_view)
        val adapter = ArrayAdapter(this,
android.R.layout.simple_list_item_1, words)
        listView.adapter = adapter

        searchView.setOnQueryTextListener(object :
SearchView.OnQueryTextListener {
            override fun onQueryTextSubmit(query:
String?): Boolean {
                return false
            }

            override fun onQueryTextChange(newText:
String?): Boolean {
                adapter.filter.filter(newText)
                return false
            }
        })
    }
}
// LessonsActivity.kt
package com.example.korean

import android.app.Activity
import android.content.Intent
import android.os.Bundle
import android.widget.AdapterView
import android.widget.AdapterView.OnItemClickListener
import android.widget.ArrayAdapter
import android.widget.ListView

class LessonsActivity : Activity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_lessons)

        val lessons = listOf("Урок 1: Знакомство с
Хангылем (корейским алфавитом) и базовыми гласными",
"Урок 2: Базовые согласные и слоги", "Урок 3:
Приветствия и базовые фразы"
, "Урок 4: Числительные (1-10)", "Урок 5:
Частицы 은 и 이/가", "Урок 6: Глаголы и основы
глаголов", "Урок 7: Настоящее время (вежливая форма)
- 오/아", "Урок 8: Прошедшее время (вежливая форма) -
았/었")
        val listView: ListView =
findViewById(R.id.lessons_list)
        val adapter = ArrayAdapter(this,
android.R.layout.simple_list_item_1, lessons)
        listView.adapter = adapter

        listView.setOnItemClickListener =
AdapterView.OnItemClickListener { _, _, position, _
->
            val intent = Intent(this,
LessonDetailActivity::class.java)
            intent.putExtra("lesson_title",
lessons[position])
            intent.putExtra("lesson_content",
getLessonContent(lessons[position]))
            startActivity(intent)
        }
    }
}

private fun getLessonContent(title: String):
String {
    return when (title) {
        else ->""
    }
}
// LessonDetailActivity.kt
package com.example.korean

import android.app.Activity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView

class LessonDetailActivity : Activity() {
    override fun onCreate(savedInstanceState:
Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(R.layout.activity_lesson_detail)

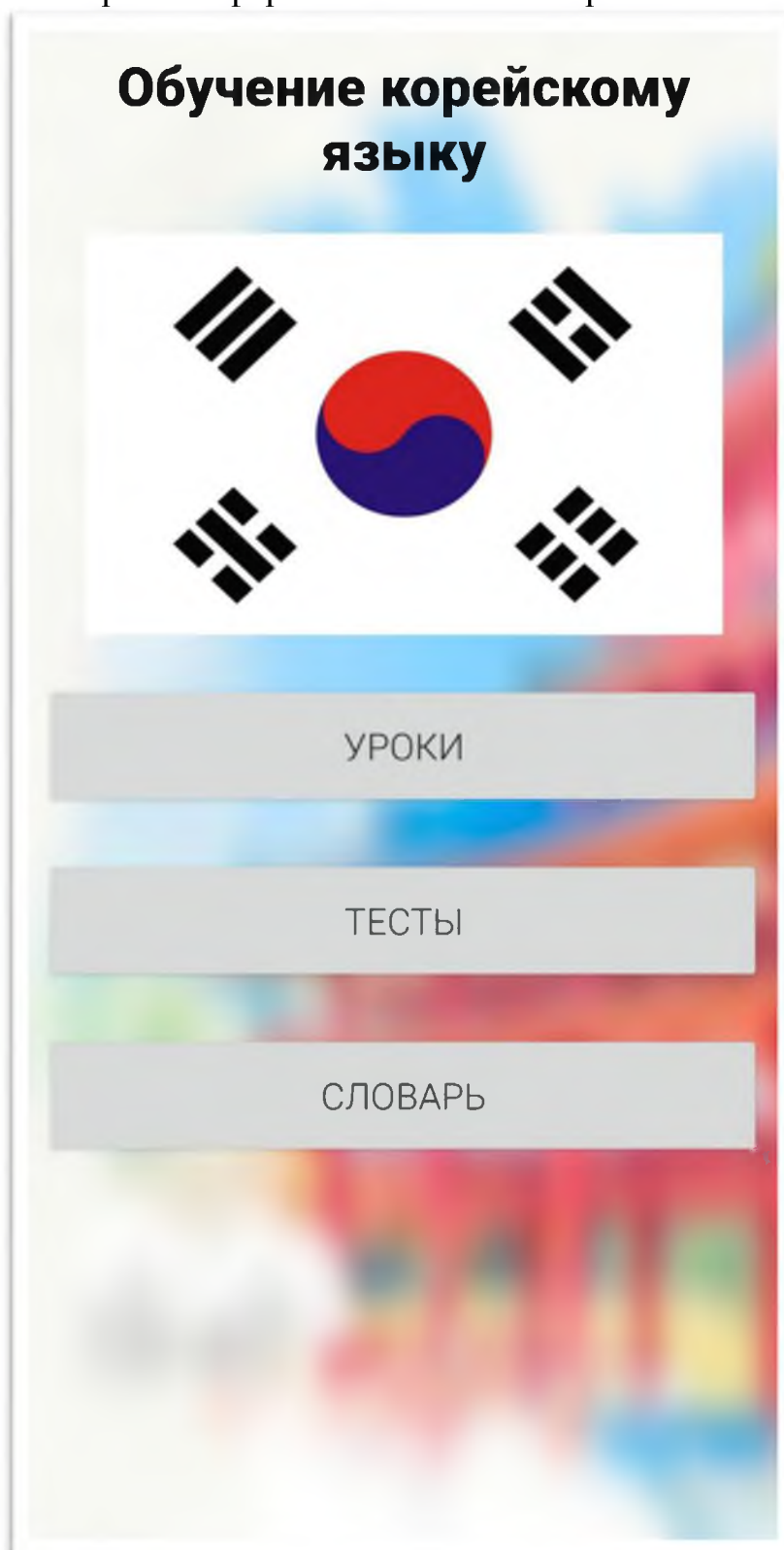
        val lessonTitle: String? =
intent.getStringExtra("lesson_title")
        val lessonContent: String? =
intent.getStringExtra("lesson_content")

        val titleTextView: TextView =
findViewById(R.id.lesson_title)
        val contentTextView: TextView =
findViewById(R.id.lesson_content)
        val backButton: Button =
findViewById(R.id.button_back)

        titleTextView.text = lessonTitle
        contentTextView.text = lessonContent
        backButton.setOnClickListener {
            finish()
        }
    }
}

```

Экранные формы. Главное окно приложения



Продолжение приложения 2

Экран «Уроки»

Урок 1: Знакомство с Хангылем (корейским алфавитом) и базовыми гласными

Урок 2: Базовые согласные и слоги

Урок 3: Приветствия и базовые фразы

Урок 4: Числительные (1-10)

Урок 5: Частицы 은/는 и 이/가

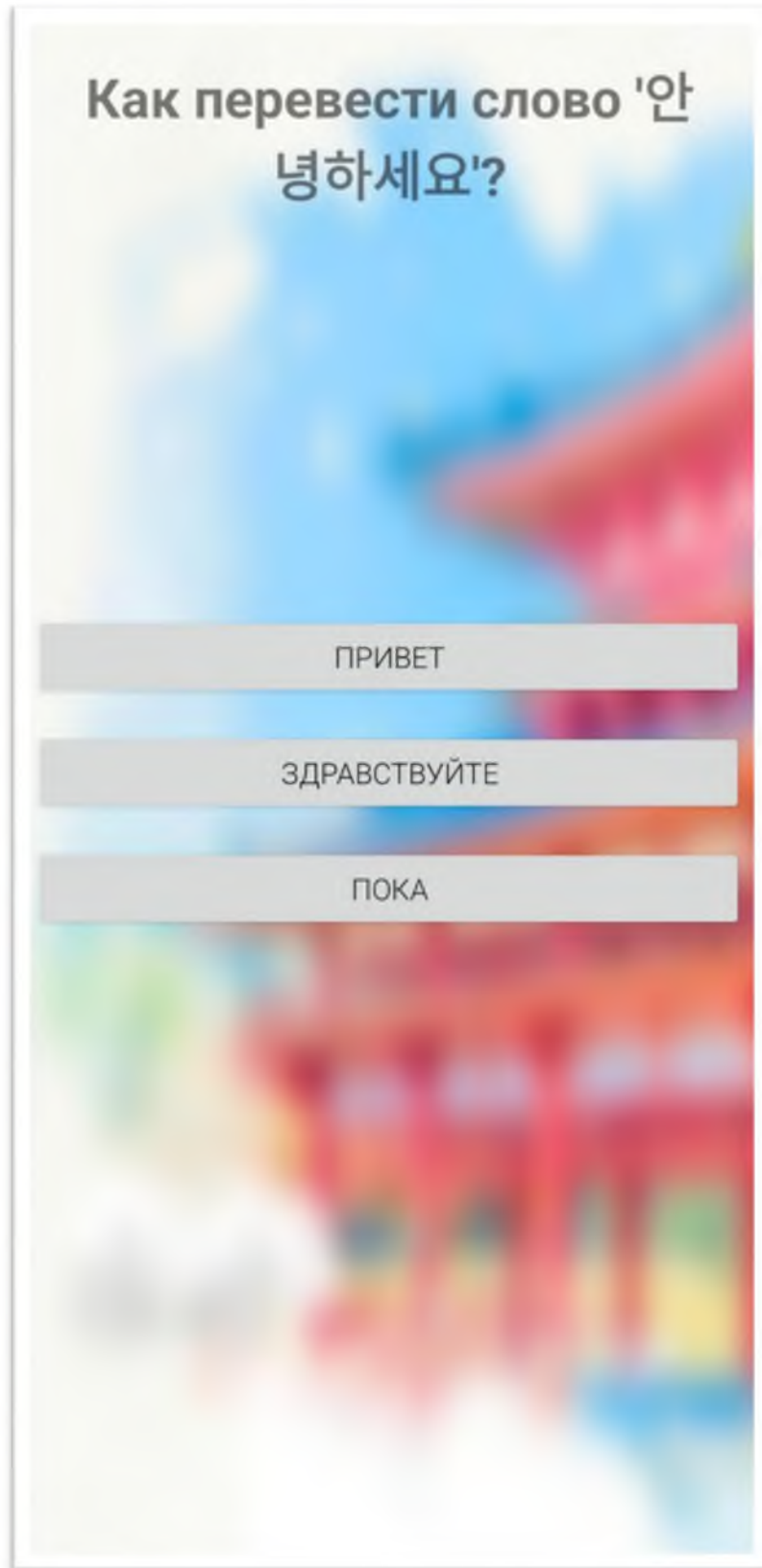
Урок 6: Глаголы и основы глаголов

Урок 7: Настоящее время (вежливая форма) -
아/어요

Урок 8: Прошедшее время (вежливая форма) -
았/었어요

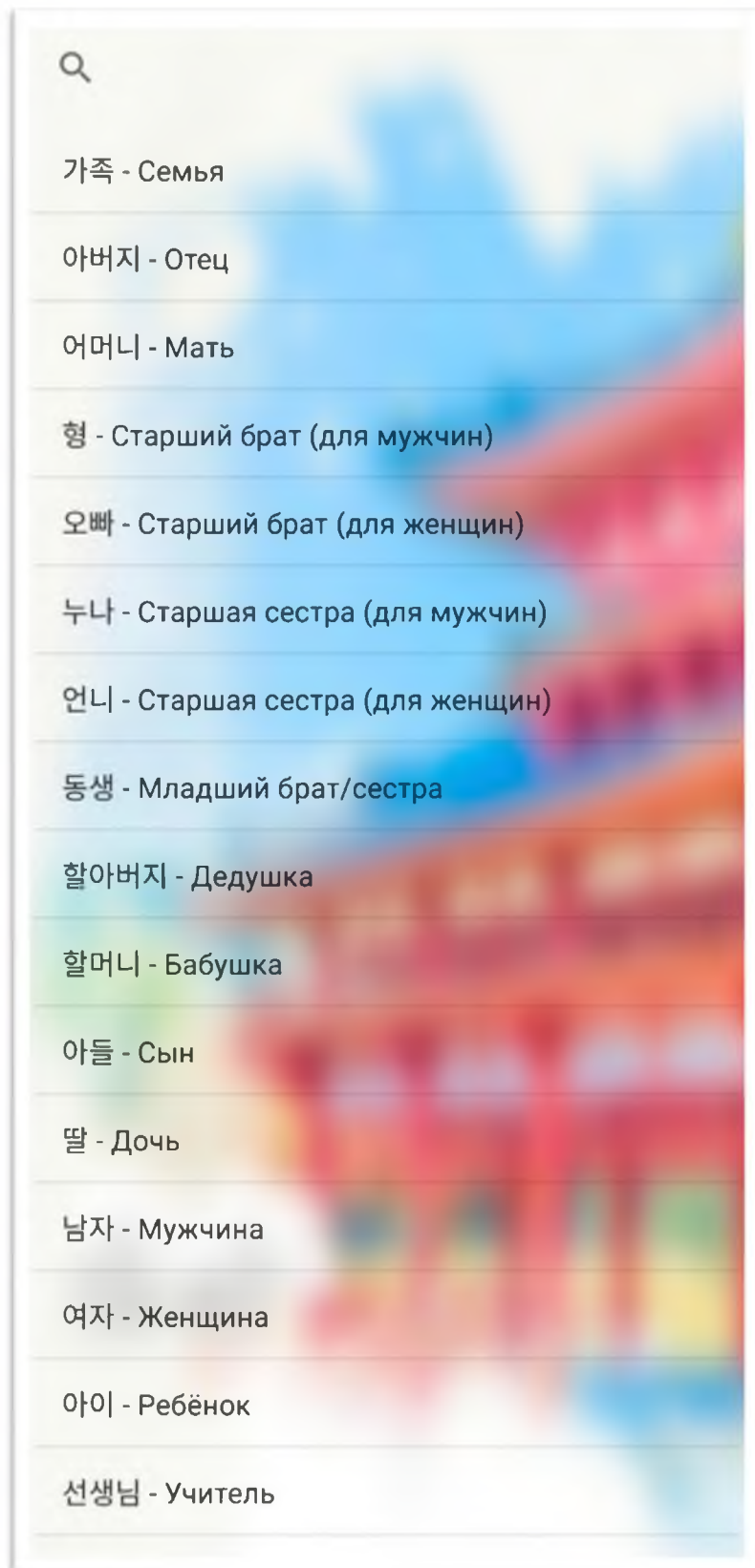
Продолжение приложения 2

Экран «Тест»



Продолжение приложения 2

Экран «Словарь»



가족	- Семья
아버지	- Отец
어머니	- Мать
형	- Старший брат (для мужчин)
오빠	- Старший брат (для женщин)
누나	- Старшая сестра (для мужчин)
언니	- Старшая сестра (для женщин)
동생	- Младший брат/сестра
할아버지	- Дедушка
할머니	- Бабушка
아들	- Сын
딸	- Дочь
남자	- Мужчина
여자	- Женщина
아이	- Ребёнок
선생님	- Учитель