

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МО и ПЭВМ

Направление подготовки 09.03.02 «Информационные системы и технологии»

БАКАЛАВРСКАЯ РАБОТА

на тему

**Мобильное приложение с дополненной реальностью для
изучения структуры растительных клеток**

Студент	_____	Гусев Дмитрий Владимирович
	(подпись, дата)	(ФИО полностью)
Руководитель	_____	Балашова И.Ю.
	(подпись, дата)	(фамилия, инициалы)
Нормоконтролёр	_____	Такташкин Д.В.
	(подпись, дата)	(фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой	_____	Макарычев П.П.
	(подпись)	(фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК	_____	Попова Н.А.
	(подпись)	(фамилия, инициалы)

Пенза, 2020

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
_____ П.П. Макарычев
«__» _____ 2020 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент Гусев Дмитрий Владимирович гр. 16ВИ1 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы Мобильное приложение с дополненной реальностью для изучения
структуры растительных клеток

Тема утверждена приказом ПГУ № 1488/о от "30" декабря 2019 г.

2. Руководитель работы к.т.н., доцент кафедры «МО и ПЭВМ» Балашова И.Ю.

3. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать приложение дополненной реальности, моделирующее строение живой
клетки для выполнения заданий по структуре клетки и процессам, происходящим в ней.
Приложение должно быть включено в образовательную систему Experiment, через
которую происходит авторизация пользователя.

Приложение должно работать на устройствах с операционной системой Android
выше седьмой версии и имеющих поддержку технологии дополненной
реальности.

Для разработки использовать среду Android Studio, программное обеспечение
NetBeans IDE и редактор 3D графики Blender.

4. Перечень подлежащих разработке вопросов

Анализ требований и постановка задачи

Проектирование

Реализация

Контроль качества ПО:

– Структурное тестирование

– Анализ качества кода

План разработки и оценка бюджета

5. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ требований и постановка задачи	10	13.04.2020 – 20.04.2020	
2	Проектирование	25	21.04.2020 – 01.05.2020	
3	Реализация	40	02.05.2020 – 29.05.2020	
4	Отладка и тестирование	15	30.05.2020 – 10.06.2020	
5	Оформление пояснительной записки	10	11.06.2020 – 13.06.2020	

Дата выдачи задания «30» декабря 2019 г.

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент _____ «30» декабря 2019 г.
(подпись, дата)

Работу к защите допустить
Декан факультета _____
(подпись, дата)

Реферат

Пояснительная записка содержит 100 листов, 46 рисунков, 22 таблицы 12 использованных источников.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИЗУЧЕНИЯ СТРУКТУРЫ РАСТИТЕЛЬНЫХ КЛЕТОК, ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ, БАЗА ДАННЫХ, POSTGRESQL, JAVA, КЛИЕНТ, СЕРВЕР, ANDROID STUDIO, DBEAVER, NET BEANS IDE.

Объект разработки – мобильное приложение с дополненной реальностью для изучения структуры растительных клеток, состоящее из базы данных, которая содержит данные о клетках и органеллах, а так же клиентского приложения для отображения 3D моделей клеток, изучения процессов проходящих в них, и получение необходимой информации.

Цель работы – разработать серверную часть мобильного приложения с дополненной реальностью для изучения структуры растительных клеток, предназначенного для его использования в образовательных учреждениях для выполнения заданий на уроках биологии.

Технология разработки – для создания серверной части используется DBEAVER и Net Beans IDE, для создания клиентской части используется среда программирования Android Studio, интерфейс для раздела тестовой части был реализована на языке JavaScript в среде программирования Web Storm с использованием фреймворка React.js.

Результаты работы – в ходе выполнения выпускной квалификационной работы было создано мобильное приложение с дополненной реальностью для изучения структуры растительных клеток. Все требования к разработке системы соблюдены.

Область применения: школы.

					ПГУ 09.03.02 – 8БР161.11 ПЗ			
Изм	Лист	№ докум.	Подпись	Дата				
Разраб.		Гусев Д.В.			Мобильное приложение с дополненной реальностью для изучения структуры растительных клеток. Пояснительная записка	Лит.	Лист	Листов
Пров.		Балашова И. Ю.					3	100
Реценз.						Гр. 16ВИ1		
Н. контр.		Такташкин Д. В.						
Утв.								

Содержание

Введение	6
1 Анализ требований и постановка задачи на разработку приложения для изучения структуры растительных клеток.....	8
1.1 Анализ рынка соответствующих продуктов.....	8
1.2 Постановка задачи на разработку серверной части мобильного приложения.....	15
1.3 Анализ функциональных требований	16
1.4 Анализ требований к пользовательскому интерфейсу приложения с дополненной реальностью для изучения структуры растительных клеток.....	23
2 Проектирование приложения по изучению структуры растительных клеток.....	26
2.1 Функциональное моделирование серверной части мобильного приложения.....	26
2.2 Проектирование структур данных.....	30
2.2.1 Логическая модель данных	31
2.2.2 Физическая модель данных	32
2.3 Проектирование программных средств	33
2.3.1 Диаграмма классов для мобильного приложения «BioCell»	34
2.3.2 Диаграмма состояний для серверной части мобильного приложения «BioCell».....	37
2.3.3 Диаграмма деятельности для серверной части мобильного приложения «BioCell».....	38
2.3.4 Диаграмма последовательности для серверной части мобильного приложения «BioCell».....	40
2.3.5 Диаграмма кооперации для мобильного приложения «BioCell».....	41
2.4 Проектирование пользовательского интерфейса	42
3 Реализация программного обеспечения.....	48
3.1 Реализация структуры данных	48

3.2 Реализация пользовательского интерфейса	52
3.3 Реализация программных средств	55
3.3.1 Диаграмма компонентов мобильного приложения для изучения структуры растительных клеток.....	55
3.3.2 Диаграмма развертывания для мобильного приложения «BioCell»...	57
4 Тестирование программного обеспечения.....	59
4.1 Структурное тестирование	59
4.1.1 Метод покрытия операторов.....	61
4.1.2 Метод покрытия условий.....	62
4.1.3 Метод покрытия маршрутов	63
4.2 Качественный анализ кода.....	64
4.2.1 SLOC-метрика	64
4.2.2 Метрика Холстеда	65
4.2.3 Цикломатическая метрика Мак-Кейба.....	67
5 Планирование и расчет бюджета разрабатываемой системы	69
Заключение	76
Список использованных источников	77
Приложение А. Глоссарий.....	79
Приложение Б. Листинг скрипта созданной базы данных на языке PostgreSQL.....	81
Приложение В. Кбспбу серверной части на языке JAVA.....	87
Приложение Г. Результаты тестирования.....	96
Приложение Д. Акт о результатах внедрения	99

Введение

В современном обществе растет роль мобильных приложений, и сфера образования не является исключением. С каждым годом появляется все больше новых развитых технологий, одной из которых является дополненная реальность. Дополненная реальность (AR/ augmented reality) – это технология, с помощью которой виртуальные объекты становятся частью реальной окружающей картины мира с помощью каких-либо устройств (смартфоны, планшеты, эмуляторы и т.п.) и специального программного обеспечения [1].

Одними из ведущих ученых в области изучения дополненной реальности являются С. Джохима, Х. Кауфманн, Л.Л. Лопез, Б. Мейер, Т. Нослони. В их работах описываются преимущества использования средств дополненной реальности в образовании, а также предлагаются учебные задания, связанные с применением данной технологии в образовательном процессе.

Использование технологии дополненной реальности в образовании позволяет обеспечить каждого обучающегося индивидуальной программой обучения и личным творческим пространством, выводя тем самым учебный процесс на более высокий уровень. Благодаря своей наглядности и интерактивности технология дополненной реальности становится все более востребованной. Вместе с этим возрастает потребность в качественном программном продукте, способном реализовать ее потенциал в полной мере.

Целью выпускной квалификационной работы является разработка серверной части мобильного приложения с дополненной реальностью для изучения структуры растительных клеток, предназначенного для использования в образовательных учреждениях на уроках биологии.

Данная работа направлена на создание приложения для изучения структуры растительной клетки на уроках биологии с применением технологии дополненной реальности «BioCell», в котором любой ученик

сможет посмотреть структуры модели клеток, узнать об их органеллах и наблюдать процессы, протекающие в них.

Применение такого приложения в учебном процессе даст возможность превратить скучный процесс обучения биологии в увлекательную игру с возможностью «прикоснуться» к клеткам и органеллам.

Актуальность темы выпускной квалификационной работы обусловлена высокой востребованностью в системе образования технологии дополненной реальности с одной стороны, и ограниченной функциональностью большинства мобильных приложений, поддерживающих данную технологию, с другой.

Для достижения указанной цели выпускной квалификационной работы были поставлены следующие задачи:

- проведение анализа предметной области и существующих мобильных приложениях аналогичного назначения;
- постановка задачи на разработку мобильного приложения с дополненной реальностью для изучения структуры растительных клеток
- определение требований к разрабатываемому мобильному приложению;
- выполнение функционального моделирования приложения;
- проектирование серверной части мобильного приложения с дополненной реальностью для изучения структуры растительных клеток;
- реализация системы;
- тестирование мобильного приложения с дополненной реальностью для изучения структуры растительных клеток.

Таким образом, применение данного мобильного приложения в учебном процессе будет способствовать повышению дидактической эффективности обучения.

Разработанные программные средства будут внедрены в международную образовательную платформу Experiment.

1 Анализ требований и постановка задачи на разработку приложения для изучения структуры растительных клеток

1.1 Анализ рынка соответствующих продуктов

Проведено исследование мобильных приложений для Android, размещенных в интернет-магазине «Play Маркет». Данная торговая площадка доступна каждому обладателю смартфона, а также она предоставляет возможность выхода разработчикам приложений на глобальный рынок с возможностью быстрого продвижения продукта [2].

Среди приложений для изучения структуры растительной клетки наиболее популярными являются: «EL-Moasser Plant Cell 3D», «iCell», «Label that Diagram - Cells», «3D Plant Cell Organelles in VR», «Cell Structure» и «Plant Cell», которые доступны бесплатно на мобильной площадке Play Маркет.

Приложение «EL-Moasser Plant Cell 3D» представлено на площадке «Play Маркет» разработчиком «GraphyMediaEG». Интерфейс приложения представлен на рисунке 1.

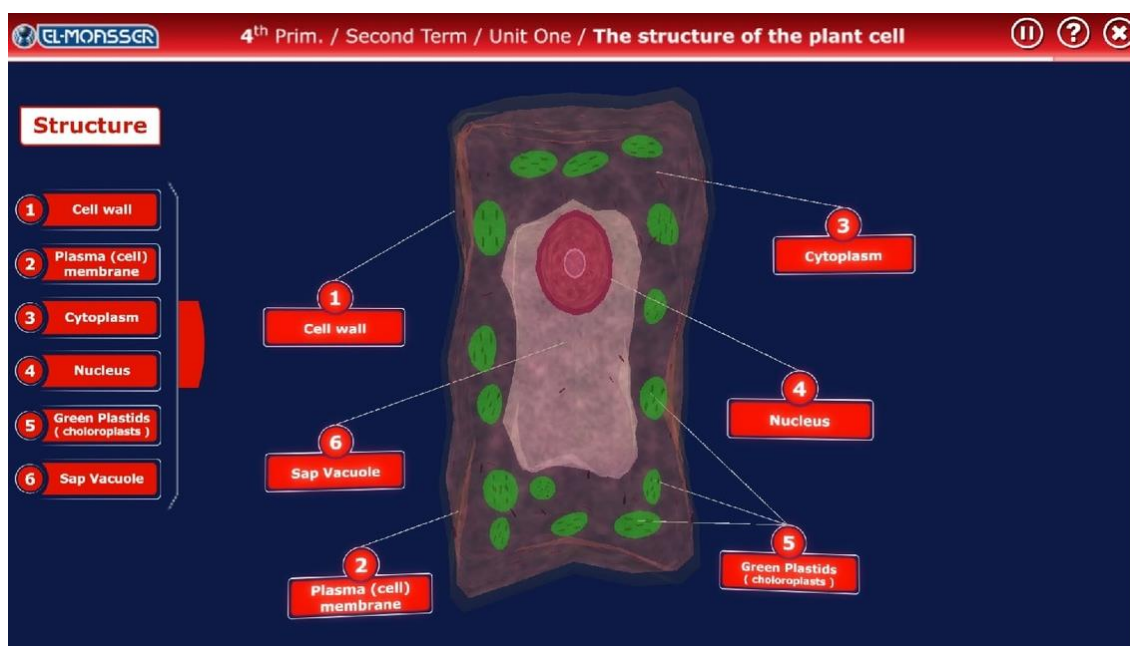


Рисунок 1 – Интерфейс приложения «EL-Moasser Plant Cell 3D»

Данное приложение позволяет просматривать структуру клеток с помощью 3D-технологий. Также оно содержит подробную информацию об органеллах и их функциях, которая воспроизводится с помощью голосового помощника. Приложение предназначено как для студентов биологических факультетов, так и учеников школ. Основным недостатком для российских учеников является то, что информация доступна только на английском языке.

Приложение «iCell» опубликовано на площадке «Play Маркет» разработчиком «HudsonAlpha Institute for Biotechnology». Интерфейс данного приложения представлен на рисунке 2.



Рисунок 2 – Интерфейс приложения «iCell»

Приложение позволяет просматривать модели нескольких видов клеток – бактерии, растительной и животной, а также внутриклеточных структур. Также в приложении дано подробное описание функциональной роли клеток и органелл.

Главным недостатком приложения является представление информации исключительно на английском языке, что ограничивает его использование в российских школах.

Приложение «Label that Diagram – Cells» представлено на площадке «Play Маркет» разработчиком «Richard Carman». Оно позволяет рассмотреть такие клетки, как растительная, животная, амёба, парамеции и эвглена. Также приложение позволяет выполнять задания по поиску органелл для их лучшего запоминания. Пользователь может выбрать один из режимов: просмотр клетки для изучения структуры или непосредственное выполнение заданий. Данный программный продукт подходит для старшеклассников средней общеобразовательной школы. Основными недостатками приложения являются:

- отсутствие подробного описания органелл и их функций;
- отсутствие 3D технологий;
- весь интерфейс приложения доступен только на английском языке.

Пример работы приложения можно увидеть на рисунке 3.

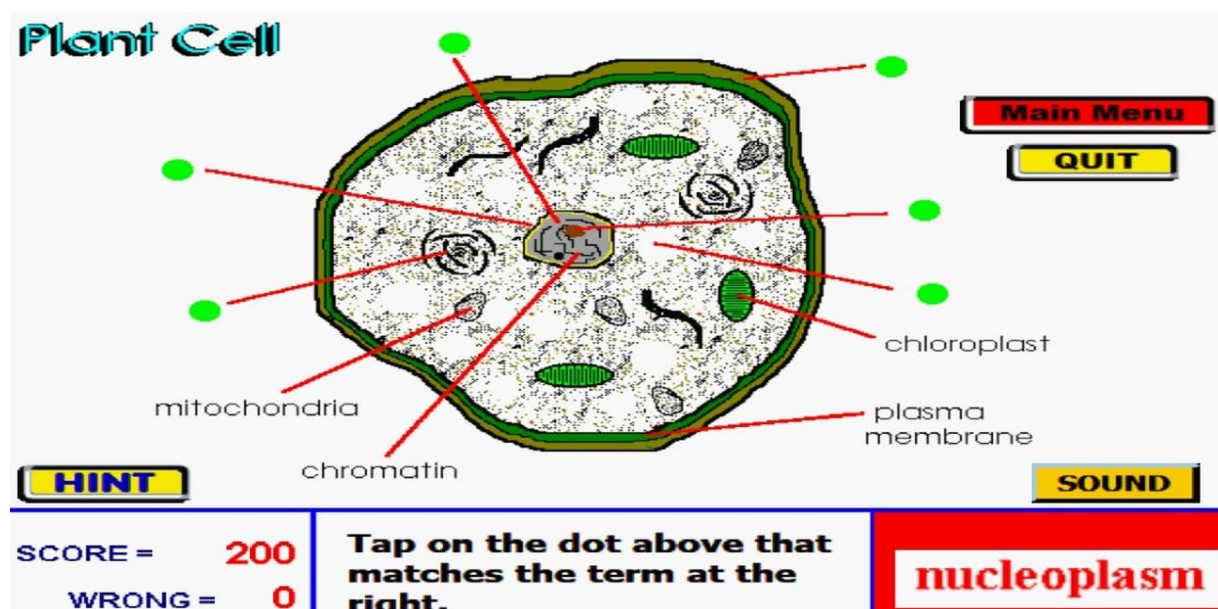


Рисунок 3 – Интерфейс приложения « Label that Diagram – Cells »

Приложение «3D Plant Cell Organelles in VR» представлено на площадке «Play Маркет» разработчиком «The Chinese University of Hong Kong». В нем можно просматривать 3D-модель с использованием очков

виртуальной реальности для смартфона. Функция виртуальной реальности позволяет взаимодействовать с виртуальными молекулами в виртуальном пространстве, что является весьма эффективным способом глубокого изучения структуры клеток.

Мобильное приложение можно использовать как школьникам общеобразовательных школ, так и студентам средне-профессиональных и высших учебных заведений. Оно очень просто в использовании, но при этом имеет довольно большой функционал. Реализована визуализация модели, как самой клеткой, так и с отдельными органеллами, кроме того, с помощью приложения можно наблюдать жизненный цикл, протекающий в клетке. В приложении реализована функция голосового помощника. К сожалению, в приложении вся информация представлена только на английском языке. Также из-за использования очков виртуальной реальности уменьшается скорость работы приложения.

Интерфейс приложения «3D Plant Cell Organelles in VR» представлен на рисунке 4.

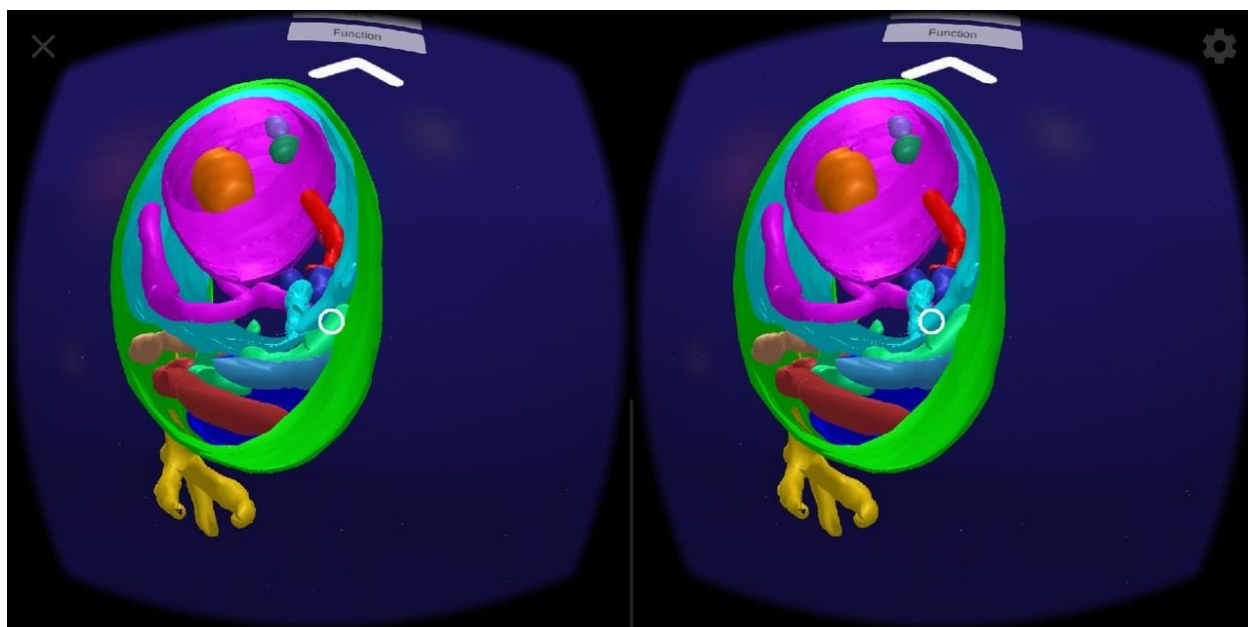


Рисунок 4 – Интерфейс приложения «3D Plant Cell Organelles in VR»

Приложение «Cell Structure» представлено на площадке «Play Маркет» разработчиком «Wiki Kids Limited». В нем можно просматривать

картинки клеток и их органелл. Мобильное приложение очень просто в использовании, и при этом имеет максимальное описание всех органелл и их функций. Так же в нем показано, как происходит транспортировка полезных веществ между органеллами в клетках, рассказывается о полезных процессах и реакциях. Основные недостатки:

- приложение полностью на английском языке;
- отсутствуют 3D-технологии.

Интерфейс приложения «Cell Structure» представлен на рисунке 5.

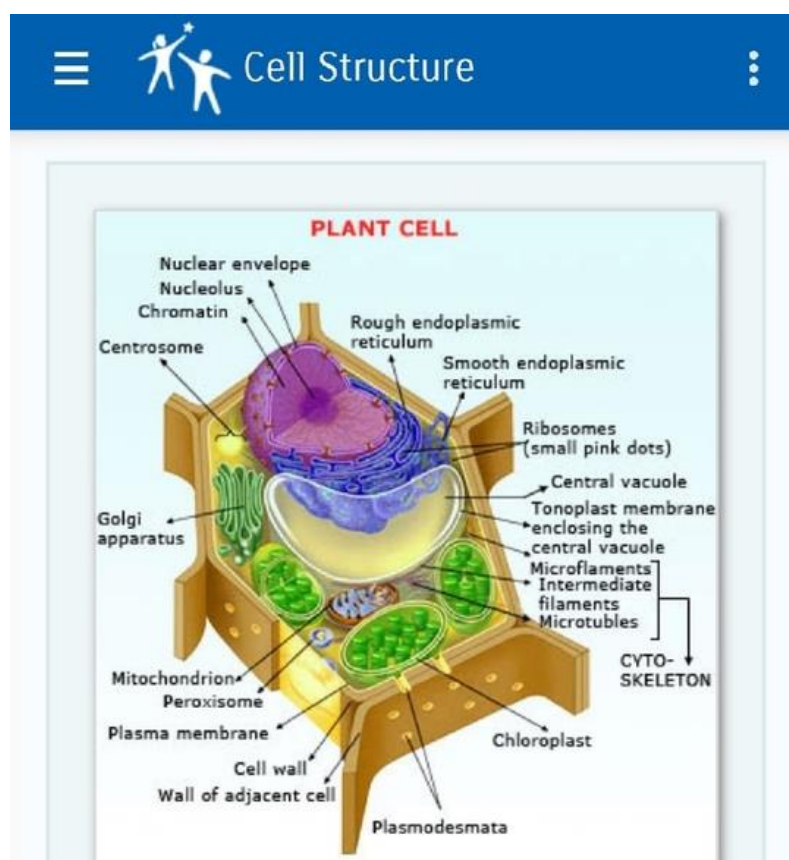


Рисунок 5 – Интерфейс приложения «Cell Structure»

Приложение «Plant Cell» представлено на площадке «Play Маркет» разработчиком «Kodingku». Интерфейс приложения состоит из обучающих видеороликов с применением 3D технологий. В данном приложении можно наблюдать за реакциями и процессами внутри клеток.

Приложение удобно в использовании, имеет дружелюбный интерфейс, имеет инструкцию по применению. Преимуществом приложения является то, что в нем можно выполнять проверочные задания,

выданные учителем. Основным недостатком приложения можно назвать то, что информация в нем представлена полностью на индонезийском языке. Интерфейс приложения «Plant Cell» представлен на рисунке 6.



Рисунок 6 – Интерфейс приложения «Plant Cell»

Рассмотренные программы обладают различным функционалом и характеристиками. В результате проведенного анализа выделены критерии, напрямую влияющие на эффективность применения приложения в учебном процессе:

- наличие описание клеток и внутриклеточных структур;
- возможность наблюдения реакций и различных процессов;
- наличие 3D-технологий;
- дополненная реальность;
- наличие банка проверочных заданий, возможность проведения автоматизированного тестирования;
- удобство приложения в использовании и скорость его работы.

Последний критерий одновременно будет устанавливать и минимальные характеристики смартфона для обработки используемой информации в приложении. Чем больше обрабатываемой информации, тем большее значение имеет скорость работы, а чем выше скорость работы приложения, тем больше ресурсов для такой обработки требуется [3].

Следовательно, минимальные характеристики смартфона будут выше среднего, т.е. это смартфоны последних двух поколений.

Полученные результаты сравнения сведены в единую таблицу (таблица 1).

Таблица 1 – Сравнение аналогов по критериям

Название	3D-технологии	Наличие AR/VR	Наблюдение реакций	Высокая скорость работы	Наличие банка проверочных заданий	Описание клеток и органелл
EL-Moasser Plant Cell 3D	+	-	-	+	-	+
iCell	+	-	-	+	-	+
Label that Diagram - Cells	-	-	-	+	+	-
3D Plant Cell Organelles in VR	+	+	+	-	-	+
Cell Structure	-	-	+	+	-	+
Plant Cell	+	-	+	+	+	+

Из результатов сравнения видно, что приложение «Plant Cell» обладает более расширенным функционалом, более удобным интерфейсом, кроме того к этому приложению разработано руководство пользователя, так же благодаря банку проверочных заданий оно позволяет организовать взаимодействие ученика и учителя.

1.2 Постановка задачи на разработку серверной части мобильного приложения

Данная выпускная квалификационная работа направлена на разработку мобильного приложения с дополненной реальностью для изучения структуры растительных клеток. Приложение бы позволило изучить модели животной клетки, растительной клетки, а также вируса и бактерии. В клиентской части приложения также должна быть реализована анимация процессов, происходящих в клетке. Приложение должно быть включено в образовательную платформу Experiment, через которую проходила бы авторизация в приложении.

Приложение должно быть разработано с помощью технологий дополненной реальности и работать на устройствах с операционной системой Android выше версии 7.0 и поддерживающих AR-технологии.

Приложение должно иметь понятный и удобный для пользователя интерфейс. Так как приложение является образовательным, то модели клеток должны быть хорошо проработаны, органеллы должны быть четко различимы, должно присутствовать название каждой органеллы и ее описание. При входе в приложение пользователю должно быть выведено сообщение о том, что в данном приложении используется технология дополненной реальности, а также уведомления о том, что приложению требуется разрешение на доступ к камере. Если пользователь не зарегистрирован в приложении Experiment Starter и не вошел в него, то должно быть показано уведомление о том, что для начала работы в приложении нужно авторизоваться.

Также для серверной части приложения должна быть реализована возможность тестирования, тесты будут создаваться учителем на сайте Experiment school, после чего задания будут отправляться в приложение.

Таким образом, можно выделить основные подзадачи на разработку мобильного приложения с дополненной реальностью для изучения структуры растительных клеток:

- изучение учебной литературы по биологии для создания моделей клеток;
- изучение дополнительной литературы для разработки приложения;
- создание моделей клеток;
- проектирование серверной части приложения;
- создание структуры приложения;
- реализация приложения дополненной реальности;
- тестирование приложения.

Использование мобильного приложения с дополненной реальностью на уроках биологии позволит:

- повысить интерес обучающихся к предмету благодаря расширению форм представления учебного контента;
- облегчить понимание учебного материала за счет повышения наглядности и интерактивности его представления;
- автоматизировать контроль знаний за счет наличия подсистемы тестирования.

Таким образом, применение данного мобильного приложения в учебном процессе будет способствовать повышению дидактической эффективности обучения.

После того, как данные задачи будут выполнены, будет создана и оформлена документация по приложению «BioCell».

1.3 Анализ функциональных требований

Приложение предназначено для изучения строения растительной клетки.

В результате анализа предметной области можно определить следующие функциональные требования:

- приложение должно использовать технологию дополненной реальности;

- приложение должно предоставлять пользователю возможность создавать объекты-клетки;
- приложение должно обеспечивать полное и корректное управление объектом-клеткой, изменять его положение в рамках рабочего пространства;
- приложение должно обеспечивать стабильную работу и высокую скорость обработки информации при проведении реакции;
- приложение должно корректно определять положение устройства в пространстве.

Для отражения функциональных требований к разрабатываемой системе была спроектирована модель в форме диаграммы вариантов использования в нотации UML. Диаграмма вариантов использования представлена на рисунке 7.



Рисунок 7 – Диаграмма вариантов использования приложения «BioCell»

В данной выпускной квалификационной работе будет рассмотрена разработка серверной части мобильного приложения с дополненной реальностью для изучения структуры растительных клеток. Диаграмма вариантов использования для серверной части представлена на рисунке 8.

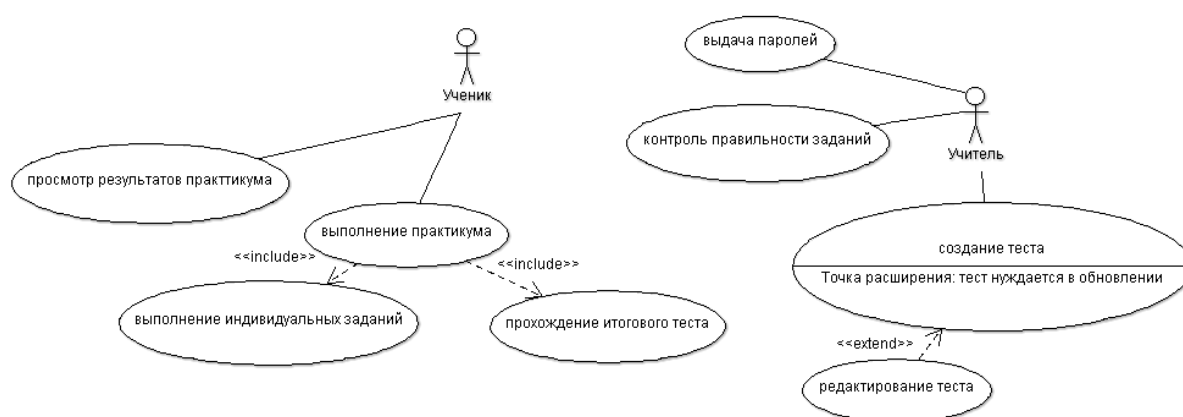


Рисунок 8 – Диаграмма вариантов использования для серверной части

Краткое описание действующего лица, изображенного на диаграмме вариантов использования приложения, представлено в таблице 2.

Таблица 2 – Описание действующего лица

Название	Профиль, подготовка и навыки
Ученик	Выполняет практические задания в приложении и получает структурированные знания о клетках, ввод ответов, получает оценку, проходит итоговый тест.
Учитель	Вносит изменения в учебный курс, создает итоговый тест, проверяет ответы, а также выдает пароли для учеников.

Для ученика можно выделить следующие варианты использования:

- выполнение практикума;
- выполнение индивидуальных заданий;
- прохождение итогового теста;
- просмотр результатов практикума.

Для учителя можно выделить следующие варианты использования:

- выдача паролей;
- создание теста;
- редактирование теста;
- контроль правильности заданий.

Краткое описание вариантов использования описано в таблице 3.

Таблица 3 – Краткое описание вариантов использования

Действующее лицо	Цель	Краткое описание
Ученик	Просмотр результатов практикума	Ученик узнает результаты выполненных заданий и тестов

Продолжение таблицы 3

Действующее лицо	Действующее лицо	Действующее лицо
Ученик	Выполнение практикума	Ученик выполняет индивидуальные задания, проходит итоговый тест, просматривает модели в свободном режиме
Ученик	Выполнение индивидуальных заданий	Ученик выполняет задания по новой теме.
Ученик	Прохождение итогового теста	Ученик проходит тест по изученному материалу
Учитель	Выдача паролей	Учитель выдает пароли для авторизации учеников
Учитель	Создание теста	Учитель создает тест для проверки знаний ученика
Учитель	Редактирование теста	Учитель вносит изменения в учебный план
Учитель	Контроль правильности заданий	Учитель проверяет результат ученика, и сравнивает с результатом компьютера, на наличие неточностей.

Полное описание вариантов использования приведено в таблицах ниже.

Описание прецедента «Выполнение практикума» приведено в таблице 4.

Таблица 4 – Описание прецедента «Выполнение практикума»

Название пункта	Описание
Наименование	Выполнение практикума
Краткое описание	Ученик выполняет определенные задания в приложении
Главные актеры	Ученик
Второстепенные актеры	Нет
Предусловие	Ученик авторизовался в приложении
Основной поток	1. Ученик выбирает режим приложения 2. Ученик приступает к изучению клетки
Постусловие	Ученик получает представления о приложении
Альтернативные потоки	- Ученик не зарегистрирован в приложении - Ученик не скачал приложение - Ученик скачивает и проходит авторизацию в приложении

После авторизации в приложение, ученик приступает к выполнению практикума. Сначала он должен выбрать модель клетки для изучения, и расположить ее в пространстве. В конце обучения у ученика присутствуют необходимые знания о клетках. Описание прецедента «Выполнение индивидуальных заданий» приведено в таблице 5.

Описание прецедента «Выполнение индивидуальных заданий» приведено в таблице 5.

Таблица 5 – Описание прецедента «Выполнение индивидуальных заданий»

Название пункта	Описание
Наименование	Выполнение индивидуальных заданий
Краткое описание	Ученик выполняет определенные задания в приложении
Главные актеры	Ученик
Второстепенные актеры	Нет
Предусловие	Ученик ознакомился с приложением
Основной поток	1. Система выдает задание 2. Ученик выбирает модель 3. Ученик выполняет задание
Постусловие	Ученик получает оценку за задания
Альтернативные потоки	- Ученик не прочел инструкции к выполнению заданий - Выдается инструкция к заданиям

После ознакомления с инструкцией для выполнения заданий, для каждого ученика, система подбирает индивидуальное знание, на основе пройденного материала и уже выполненных заданий.

Описание прецедента «Прохождение итогового теста» приведено в таблице 6.

Таблица 6 – Описание прецедента «Прохождение итогового теста»

Название пункта	Описание
Наименование	Прохождение итогового теста
Краткое описание	Ученик проходит итоговый тест
Главные актеры	Ученик
Второстепенные актеры	Нет
Предусловие	Ученик выполнил практические задания
Основной поток	1. Выдается список вопросов 2. Ученик выполняет тест 3. Ученик проверяет результаты
Постусловие	Ученик получает оценку за тест
Альтернативные потоки	- Ученик не выполнил практические задания - Возврат к выполнению заданий

Когда ученик выполнил достаточное количество заданий, по определенной теме, ему выдается тест. Тест создается учителем на специальном сайте. После его прохождения, учитель анализирует результаты и выставляет оценку.

Описание прецедента «Просмотр результатов практикума» приведено в таблице 7.

Таблица 7 – Описание прецедента «Просмотр результатов практикума»

Название пункта	Описание
Наименование	Просмотр результатов практикума
Краткое описание	Ученик смотрит результаты за задания и тесты
Главные актеры	Ученик
Второстепенные актеры	Учитель
Предусловие	Ученик выполнил работу
Основной поток	Ученик проверяет оценку за выполненную работу
Постусловие	Подсистема выводит данные об оценках
Альтернативные потоки	<ul style="list-style-type: none"> - Задания еще не выполнены - Задания еще не проверены - Возврат в меню

После выполнения практикума, ученик сразу может проверить свои результаты. Будут выведены правильные ответы, и ответы выбранные учеником, где он сможет узнать, какую тему ему необходимо подтянуть.

Описание прецедента «Контроль правильности заданий» приведено в таблице 8.

Таблица 8 – Описание прецедента «Контроль правильности заданий»

Название пункта	Описание
Наименование	Контроль правильности заданий
Краткое описание	Учитель проверяет работу ученика
Главные актеры	Учитель
Второстепенные актеры	Нет
Предусловие	Ученик выполнил работу
Основной поток	Учитель проверяет результат ученика, и сравнивает с результатом компьютера, на наличие неточностей
Постусловие	Подсистема выводит данные о результате работы ученика
Альтернативные потоки	<ul style="list-style-type: none"> - Задания еще не выполнены - Задания еще не проверены - Возврат в меню

После выполнения теста учеником, его результаты отображаются на сайте. Учитель может посмотреть, сколько баллов набрал ученик из их максимального количества, и выставить соответствующую оценку.

Описание прецедента «Выдача паролей» приведено в таблице 9.

Таблица 9 – Описание прецедента «Выдача паролей»

Название пункта	Описание
Наименование	Выдача паролей
Краткое описание	Учитель выдает пароли ученикам
Главные актеры	Учитель
Второстепенные актеры	Нет
Предусловие	Программист зарегистрировал учеников
Основной поток	Учитель выдает пароли
Постусловие	Ученик получает пароль
Альтернативные потоки	- Ученик еще не зарегистрирован - Программист регистрирует ученика

Программист регистрирует учеников школы, после чего выдает данные об учетной записи классному руководителю. Для доступа к приложению, учитель выдает необходимые данные ученикам.

Описание прецедента «Создание теста» приведено в таблице 10.

Таблица 10 – Описание прецедента «Создание теста»

Название пункта	Описание
Наименование	Создание теста
Краткое описание	Учитель создает новый тест
Главные актеры	Программист
Второстепенные актеры	Нет
Предусловие	Пройден новый материал
Основной поток	Учитель добавляет тест на сайте
Постусловие	Обновленное приложение
Альтернативные потоки	Нет

После пройденного материала, учитель создает тест на сайте. Для теста необходимо выбрать его тему, класс, сколько разрешено попыток выполнения, и время прохождения. К каждому тесту можно добавлять и изменять вопросы, для которых учитель придумывает задание, сложность задания и правильный ответ.

Описание прецедента «Редактирование теста» приведено в таблице 11.

Таблица 11 – Описание прецедента «Редактирование теста»

Название пункта	Описание
Наименование	Редактирование теста
Краткое описание	Учитель вносит изменения в учебный план
Главные актеры	Учитель

Продолжение таблицы 11

Название пункта	Описание
Второстепенные актеры	Программист
Предусловие	Неточности в учебном плане
Основной поток	1. Учитель вносит изменения в тест 2. Программист вносит изменения в приложение 3. Программист обновляет приложение
Постусловие	Обновленное приложение
Альтернативные потоки	Нет

У учителя есть возможность отредактировать тест, изменив его формулировку, или изменить структуру вопросов.

Таким образом, функциональные требования были объединены и изображены графически с помощью диаграммы вариантов использования.

1.4 Анализ требований к пользовательскому интерфейсу приложения с дополненной реальностью для изучения структуры растительных клеток

Требования к пользовательскому интерфейсу являются одними из основных, которые определяют частично функционал, а частично внешний вид приложения [4]. При анализе принципов дизайна материального интерфейса были выведены следующие требования к пользовательскому интерфейсу приложения с дополненной реальностью для изучения структуры растительных клеток:

- обязательное размещение рабочей области;
- объекты должны соответствовать физическим ограничениям, действующим в реальном мире;
- способность поддерживать параллельную деятельность, т.е. несколько объектов и элементов интерфейса могут манипулировать одновременно;
- обязательная поддержка физического взаимодействия объектов, т.е. использование пространственных отношений.

Наряду с чисто функциональными требованиями к интерфейсу, у приложения существуют требования дизайна:

- наличие развернутого пользовательского меню (система подсказок и т.п.);
- текстовое меню должно быть написано на доступном пользователю языке;
- наличие описания для клеток и органелл;
- наличие визуального представления клетки в виде сферы с обозначениями.

1.5 Анализ требований к средствам разработки приложения для изучения структуры растительных клеток

На основе изучения различных средств для разработки приложения с технологией виртуальной и дополненной реальности был выведен ряд требований:

- возможность бесплатно установить из сети Интернет;
- библиотека 3D объектов должна иметь возможность работать с современными форматами трехмерной графики – *.3ds, *.dae, *.fbx, *.flt;
- поддержка языков программирования высокого уровня;
- наличие редактора программной и графической разработки объектов;
- возможность подключения сторонних библиотек объектов.

Исходя из перечисленных требований в качестве движка был выбран Android Studio, который доступен бесплатно для некоммерческих целей. Для управления базы данных используется программа DBeaver, работающая с языком PostgreSQL. В качестве расширения функционала и подключения дополнительных библиотек используется Net Beans IDE. Для связывания серверной части и базы данных используется программное обеспечение Web Storm. Для проектирования приложения были использованы CASE-средство ERwin Data Modeler 7.3 и ArgoUML. Для

проектирования структур данных была использована нотация IDEF1X, а для проектирования программных средств использовался унифицированный язык моделирования UML.

Для планирования проекта и расчета бюджета был использован Microsoft Project, который позволил внести используемые ресурсы, рассчитать их использование и организовать продуктивную и своевременную работу над проектом.

Для разработки 3D объектов, которые используются в приложении, был выбран пакет для создания трехмерной компьютерной графики Blender. Основными его достоинствами служат бесплатный доступ к полному его функционалу, простота использования и большое количество инструкций [5].

2 Проектирование приложения по изучению структуры растительных клеток

2.1 Функциональное моделирование серверной части мобильного приложения

Для проектируемого мобильного приложения с дополненной реальностью для изучения структуры растительных клеток были выделены бизнес-процессы, которые представлены в виде диаграмм при помощи CASE-средства CA ERwin Process Modeler, поддерживающего такие методологии моделирования бизнес-процессов, как IDEF0, IDEF3 и DFD. CA ERwin Process Modeler является средством визуального моделирования

IDEF0 используется для создания функциональной модели, отображающей структуру и функции приложения, а также потоки информации и материальных объектов, связывающие эти функции [6]. Контекстная диаграмма бизнес-процессов проектируемой системы в нотации IDEF0 представлена на рисунке 9.

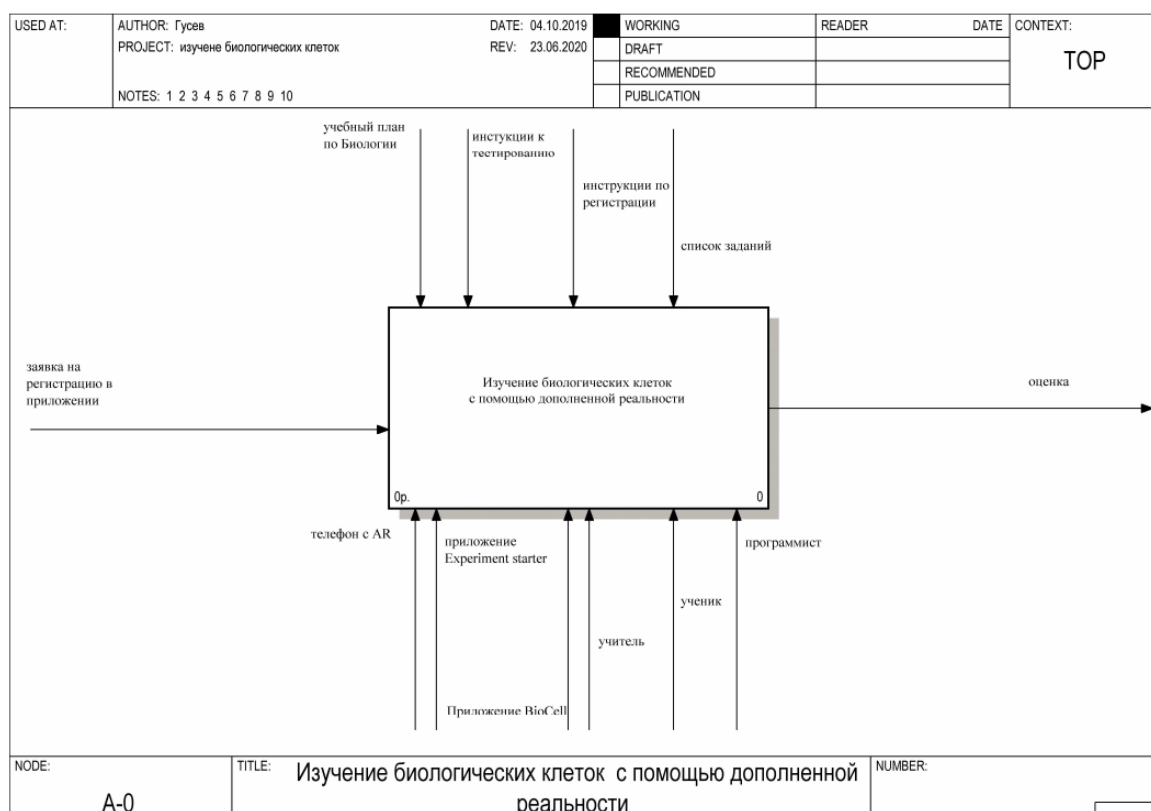


Рисунок 9 – Контекстная диаграмма приложения для изучения биологических клеток в нотации IDEF0

Диаграммы IDEF0 позволяют понять, какие объекты или информация служат ресурсами для процессов, какие результаты производят работы, что является управляющими факторами и какие ресурсы для этого необходимы.

На контекстной диаграмме отражается общее описание основного бизнес-процесса (блока) системы – изучение биологических клеток с помощью дополненной реальности.

После описания системы проводится ее разбиение на фрагменты. Этот процесс называется функциональной декомпозицией [6]. Основной бизнес-процесс (изучение биологических клеток с помощью дополненной реальности) был декомпозирован на следующие составляющие бизнес-процессы: регистрация в Experiment, предоставление учебного материала по биологическим клеткам, выполнение тестовых заданий в приложении, прохождение теста. Диаграмма декомпозиции представлена на рисунке 10.

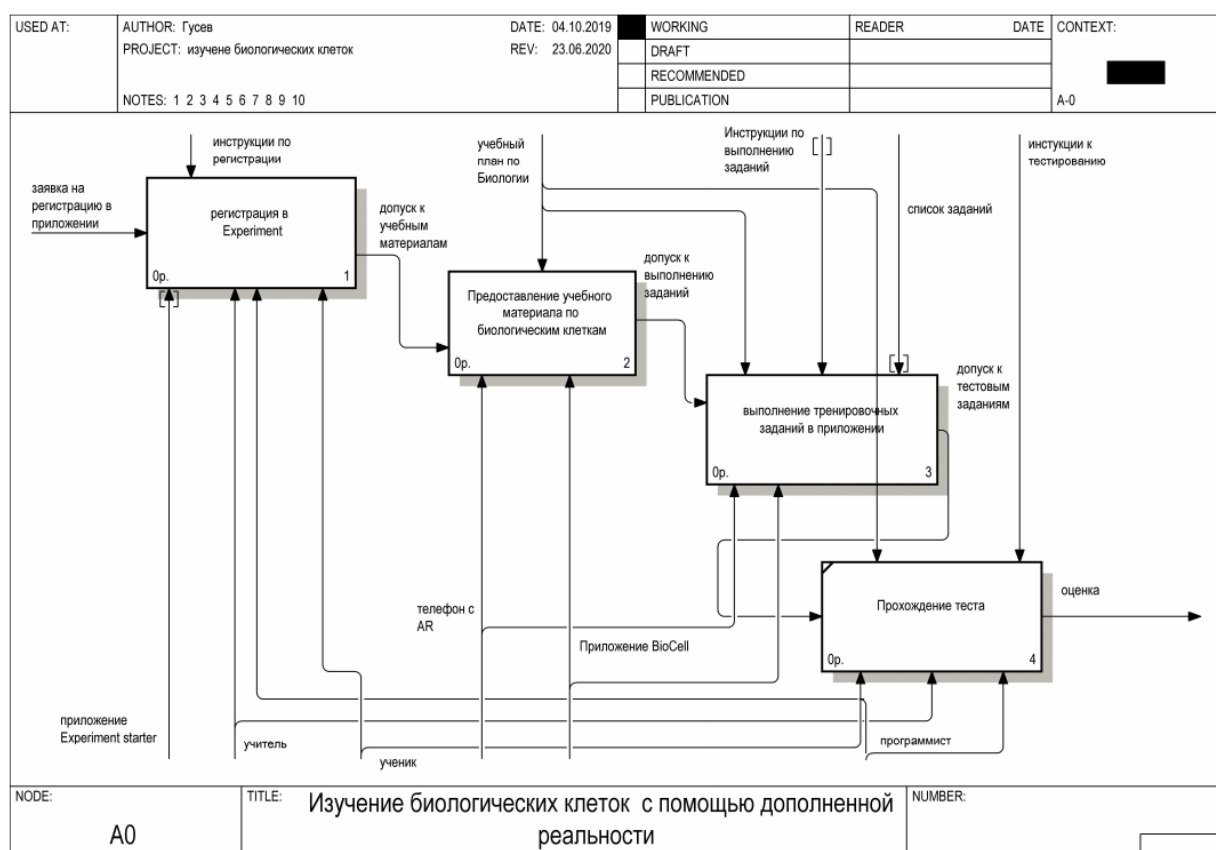


Рисунок 10 – Диаграмма декомпозиции приложения для изучения биологических клеток в нотации IDEF0

Программист сначала регистрирует пользователей в приложении Experiment, после чего пользователь может авторизоваться и приступить к

выполнению заданий. Для проверки своих знаний, ученик получает задания и выполняет их в самом приложении. Для закрепления пройденного материала, ученикам выдаются тесты, после которых они приобретают знания и получают оценку.

Функциональные блоки на диаграмме декомпозиции изучения биологических клеток также будут декомпозированы. Бизнес-процесс регистрации в Experiment декомпозирован в нотации IDEF0. Диаграмма декомпозиции представлена на рисунке 11.

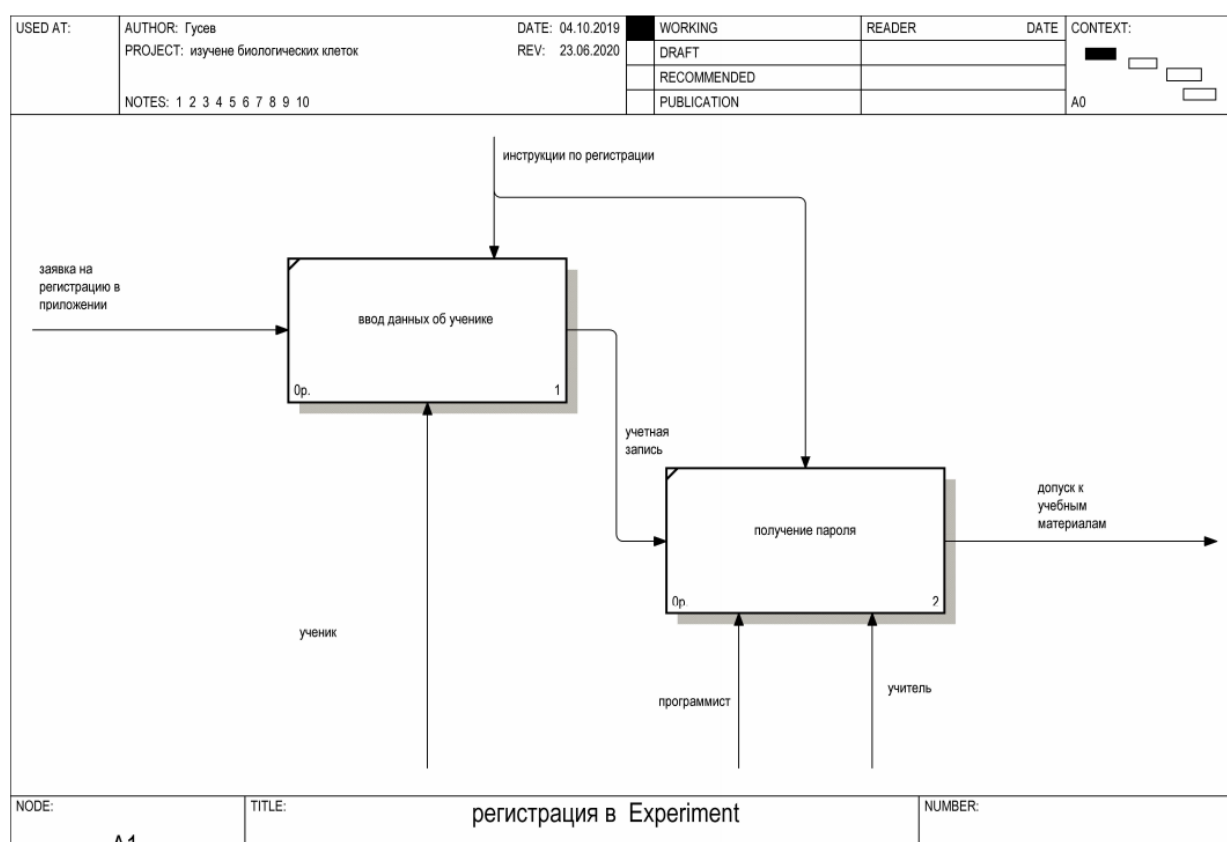


Рисунок 11 – Диаграмма декомпозиции «Регистрация в Experiment» в нотации IDEF0

Для регистрации ученик вводит всю необходимую информацию на сайте Experiment School, после чего программист создает учетную запись ученика и передает данные для входа учителю.

Бизнес-процесс «Предоставление учебного материала по биологическим клеткам» декомпозирован в нотации IDEF3. IDEF3 является нотацией для описания логики взаимодействия информационных потоков.

Декомпозиция бизнес-процесса «Предоставление учебного материала по биологическим клеткам» представлена на рисунке 12.

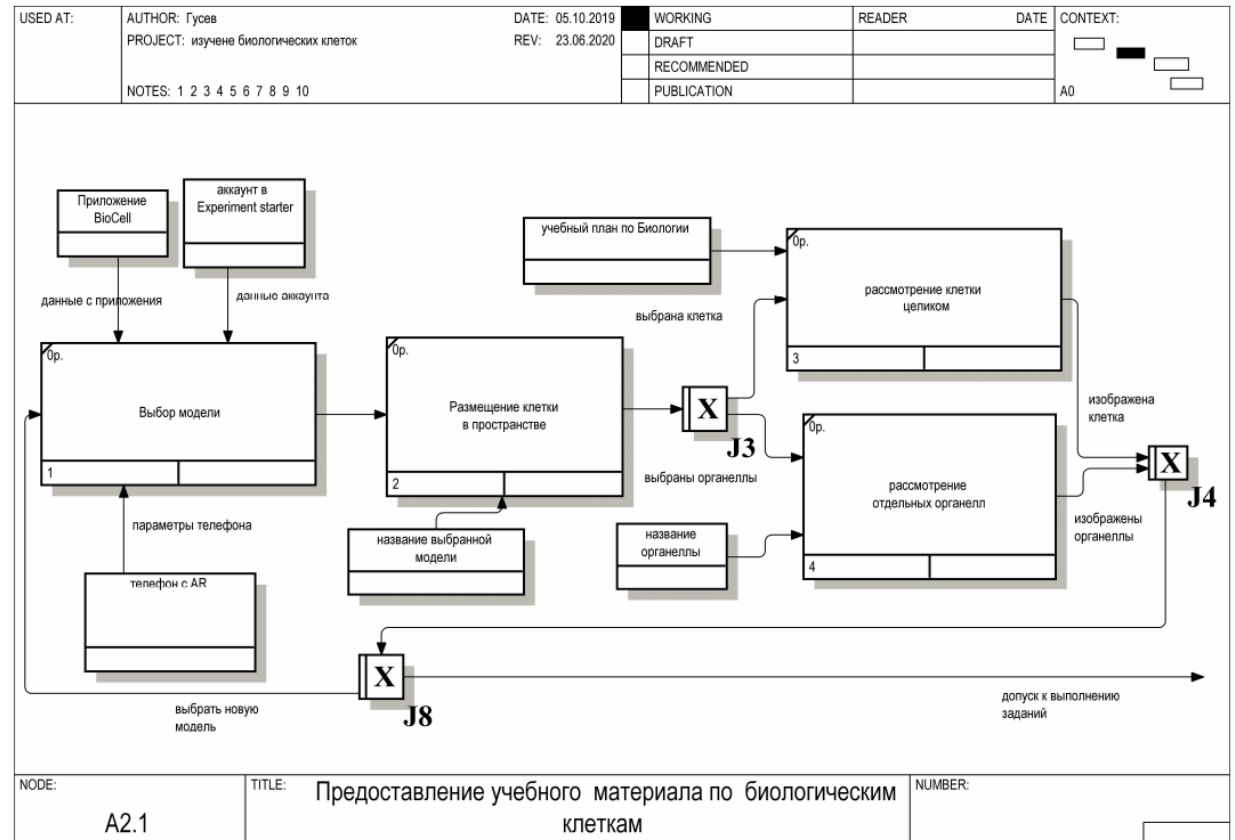


Рисунок 12 – Декомпозиция бизнес-процесса «Предоставление учебного материала по биологическим клеткам» в нотации IDEF3

Для начала работы пользователю необходимо выбрать модель клетки, с которой он хочет работать. После чего необходимо разместить 3D объект модели клетки в пространстве. Так же у пользователя есть возможность выбрать для рассмотрения всю клетку целиком, или рассмотреть отдельно ее органеллы.

Бизнес-процесс «Прохождение теста» декомпозирован в нотации DFD (Data Flow Diagram). Диаграммы потоков данных позволяют описать документооборот и обработку данных. В нотации DFD хранилище позволяет описать и сохранить в памяти данные. Внешние сущности представляют собой источник получения данных. Стрелки описывают движение объектов из одной части системы в другую [6]. Декомпозиция бизнес-процесса «Прохождение теста» представлена на рисунке 13.

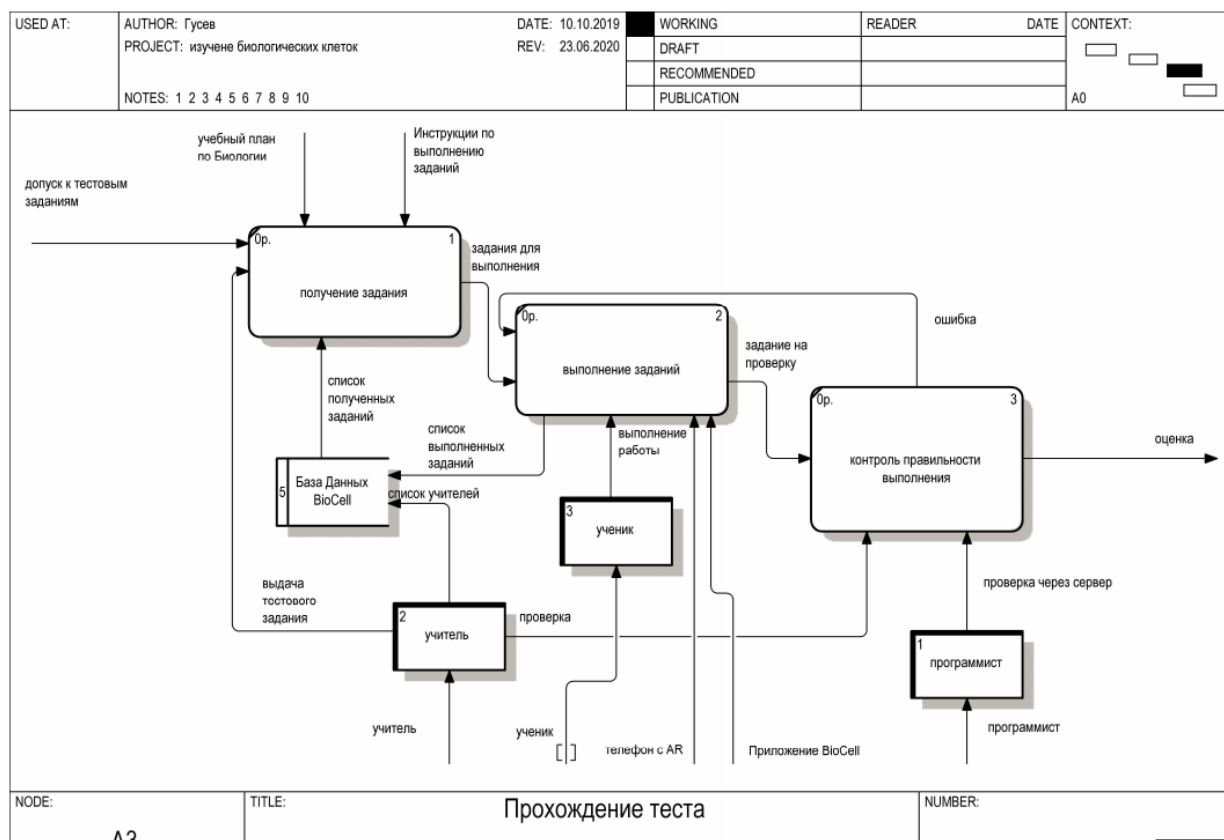


Рисунок 13 – Декомпозиция бизнес-процесса «Прохождение теста» в нотации DFD

После ознакомления с приложением, ученику необходимо получить тестовые задания и ознакомиться с инструкциями для его выполнения. Выполнив все необходимые задания, у ученика есть возможность проверить правильность их выполнения. Правильность выполненных заданий учеником проверяется в самом приложении автоматически, учитель может посмотреть результаты прохождения тестов на сайте Experiment School.

2.2 Проектирование структур данных

Проектирование структуры данных осуществлялось с помощью Case-средства ERwin Data Modeler 7.3 и его методологии IDEF1X. Благодаря простоте интерфейса, а также поддержке большого количества платформ и специфических особенностей организации физической памяти различных серверных СУБД ERwin является наиболее известным и доступным средством моделирования данных.

Для создания структуры данных в стандарте IDEF1X были рассмотрены два уровня представления модели данных – логический и физический.

2.2.1 Логическая модель данных

На основе проведенного анализа предметной области и функциональных требований к системе была построена модель «сущность-связь» для базы данных на логическом уровне. Логический уровень – это абстрактный взгляд на данные, в котором данные представлены также, как и в реальном мире. Логическая модель включает сущности, атрибуты и связи, а также полностью отражает суть выбранной предметной области.

Проектирование логической модели БД было произведено с помощью CASE-средства моделирования данных CA ERwin Data Modeler в нотации IDEF1X.

Построенная логическая модель в нотации IDEF1X представлена на рисунке 14.

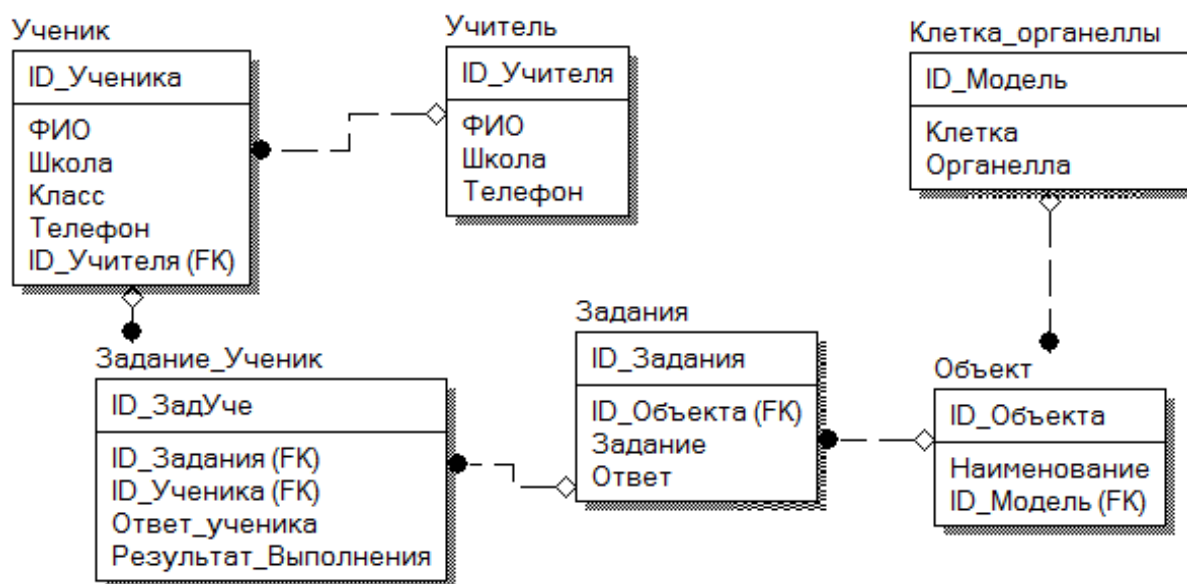


Рисунок 14 – Логическая модель базы данных

ERwin автоматически поддерживает согласованность логической и физической схем модели данных и обеспечивает автоматическую генерацию файлов (БД) в различных форматах [7], в том числе и в Microsoft

SQL Server. Методология IDEF1X разработана специально для проектирования реляционных БД.

В дальнейшем будет произведен переход от логического уровня модели БД к физическому уровню.

2.2.2 Физическая модель данных

На основе логической модели БД была построена физическая модель на атрибутивном уровне представления сущностей с полным перечнем атрибутов, ключей и указанием типов данных, которыми описаны атрибуты. Так как физическая модель зависит от конкретной реализации СУБД, то в ней важно полностью описать информацию обо всех объектах БД. Реализованная модель учитывает особенности целевой СУБД Microsoft SQL Server.

Физическая модель, изображенная на рисунке 15, была построена в среде Erwin Data Modeler в нотации IDEF1X и полностью описывает сущности предметной области.

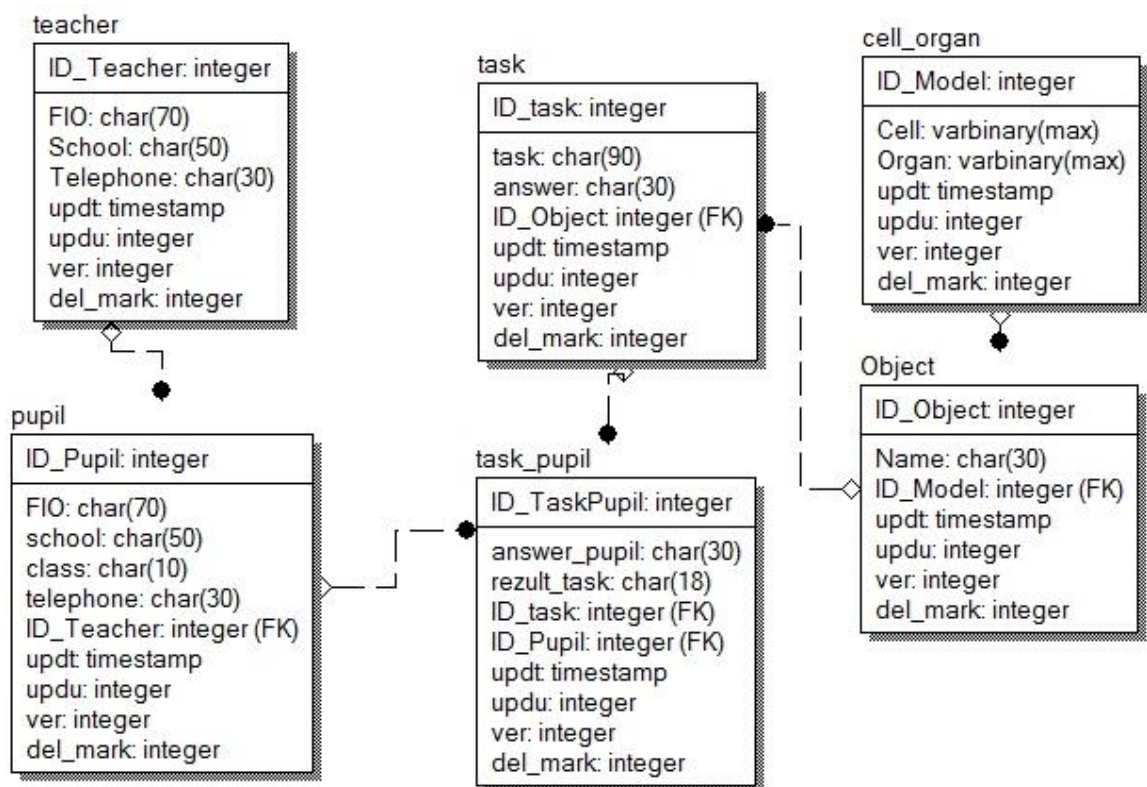


Рисунок 15 – Физическая модель базы данных

В физической модели были добавлены типы данных атрибутов, а также были добавлены такие атрибуты, как:

- updt – атрибут, который нужен для определения даты и времени обновления базы, необходим в ситуациях, когда нужно отменить произведенные изменения;
- updu – атрибут, сохраняющий ID пользователя, который обновлял базу последним;
- ver – версия обновления;
- del_mark – пометка на удаление.

2.3 Проектирование программных средств

Для проектирования программных средств для приложения по изучению структуры растительных клеток был выбран объектно-ориентированный подход. Объектный подход представляет моделируемую систему как набор взаимодействующих объектов. Объект обладает конкретно определяемым поведением. Целью использования данной методологии является акцентирование объектов, образующих систему, и распределение среди них ответственностей за производимые действия.

В рамках объектного подхода были спроектированы диаграммы с помощью CASE-средства UML-моделирования ArgoUML. UML является языком визуального моделирования, который базируется на объектно-направленном подходе. Главная концепция UML – возможность моделировать системы как наборы взаимодействующих объектов. Диаграмма в UML – это графическое представление набора элементов.

Для этапа проектирования в методологии UML были выбраны следующие виды поведенческих диаграмм: диаграмма классов, диаграмма состояний, диаграмма деятельности, диаграмма последовательности, диаграмма кооперации.

2.3.1 Диаграмма классов для мобильного приложения «BioCell»

Диаграмма классов описывает типы классов системы и разнообразные связи, которые имеются между ними. Кроме того, в диаграммах классов представляются атрибуты и методы классов. Атрибуты представляют свойства объектов класса. За именованим атрибута может следовать его тип. Метод представляет функцию класса. Метод способен обладать параметрами и возвращать значения.

На основе анализа предметной области была построена диаграмма программных классов, представленная на рисунке 16.

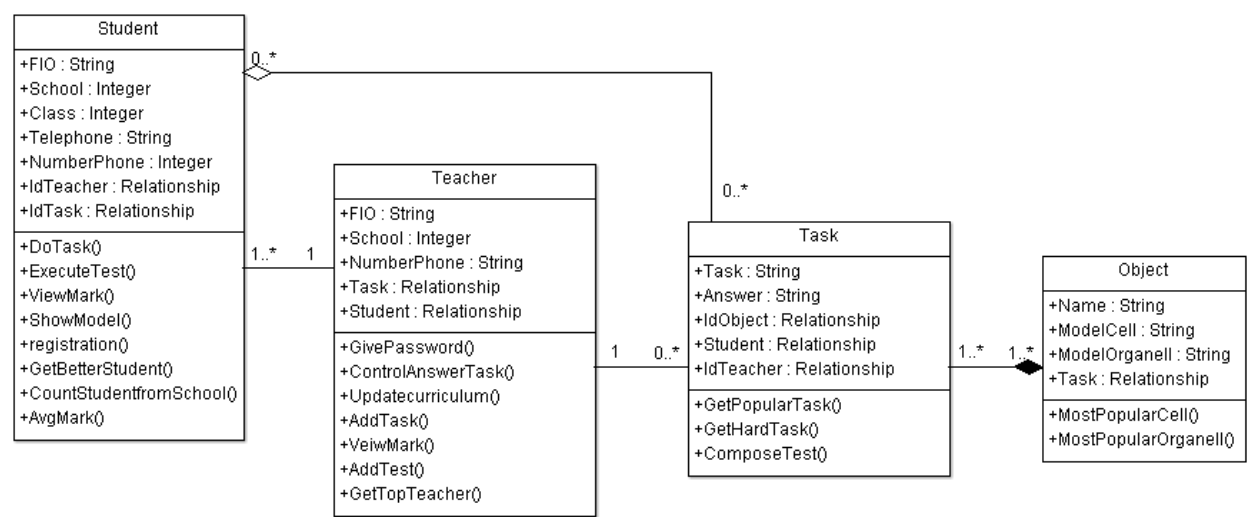


Рисунок 16 – Диаграмма классов приложения с дополненной реальностью для изучения структуры растительных клеток

Полное описание представленных классов приведено в таблицах ниже. Описание класса Object представлено в таблице 12.

Таблица 12 – Описание класса Object

Характеристики	Описание
Имя класса:	Object
Назначение:	Описывает список объектов
Атрибуты:	Name: String(70) – Название объекта ModelCell: String(40) – модель клетки ModelOrganell: String(40) – модель органеллы Task : Relationship – связь с классом задание
Методы:	MostPopularCell() – получить самую популярную клетку MostPopularOrganell() – получить самую популярную органеллу
Связь:	Task - связь композиция. Задание имеет привязку к объекту, которое имеет жесткую зависимость времени существования экземпляра класса Объект. Если будет удален объект, то и удалится задание.

В классе Object хранятся данные о том, как связаны между собой клетки и органеллы, и так же присутствует связь между заданиями и внутриклеточными структурами. Так же прописывается путь, где хранятся модели объектов.

Описание класса Teacher представлено в таблице 13.

Таблица 13 – Описание класса Teacher

Характеристики	Описание
Имя класса:	Teacher
Назначение:	Описывает учителя.
Атрибуты:	FIO: String (70) – фамилия имя отчество учителя. School: String(35) – школа учителя NumberPhone: String(35) – номер телефона учителя Student : Relationship – Связь с классом студент Task: Relationship – Связь с классом задание
Методы:	GivePassword() – выдача пароля ControlAnswerTask() – проверка ответа ученика на задание UpdateCurriculum() – обновление учебного плана AddTask() – добавление задания AddTest() – добавление теста ViewMark() – просмотреть оценки GetTopTeacher() – список лучших учителей
Связь:	Student - связь Ассоциация. Ассоциация показывает, что объекты класса Teacher связаны с объектами класса Student таким образом, что можно перемещаться от объектов одного класса к другому. Task - связь Ассоциация. Ассоциация показывает, что объекты класса Teacher связаны с объектами класса Task таким образом, что можно перемещаться от объектов одного класса к другому.

В классе Teacher хранятся все данные об учителях, работающих с приложением, такие как ФИО, номер телефона, школа, классы, у которых они ведут занятие. Учитель может добавлять и изменять тест и вопросы к ним, кроме того учитель проверяет выполненные задания учеников и выставляет им оценки. Так же учитель может изменять учебный план, в зависимости от пройденного курса.

Описание класса Task представлено в таблице 14.

Таблица 14 – Описание класса Task

Характеристики	Описание
Имя класса:	Task
Назначение:	Описывает список заданий.
Атрибуты:	Task: String(70) – Описание задания Answer: String(50) – ответ на задание IdObject : Relationship – Связь с классом объект Student : Relationship – Связь с классом студент IdTeacher : Relationship – Связь с классом учитель
Методы:	GetPopularTask() – Вывод списка популярных заданий GetHardTask() – Вывод списка сложнее заданий ComposeTask() – Составление теста
Связь:	Student - связь Агрегация. Класс Student включает в себя несколько заданий. При добавлении задания, происходит его привязка к ученику Object – связь композиция. Задание имеет привязку к объекту, которое имеет жесткую зависимость времени существования экземпляра класса Объект. Если будет удален объект, то и удалится задание. Teacher - связь Ассоциация. Ассоциация показывает, что объекты класса Task связаны с объектами класса Teacher таким образом, что можно перемещаться от объектов одного класса к другому.

В классе Task хранится вся необходимая информация о заданиях, это постановка вопроса, ответ на него и связь, какому ученику было выдано это задание.

Описание класса Student представлено в таблице 15.

Таблица 15 – Описание класса Student

Характеристики	Описание
Имя класса:	Student
Назначение:	Описывает ученика, обучающегося в приложении.
Атрибуты:	FIO: String(70) – фамилия имя отчество студента. School: String(35) – школа ученика Class: String(10) – класс ученика. Telephone: String(30) – модель телефона ученика NumberPhone: String(35) – номер телефона ученика IdTeacher : Relationship – Связь с классом учитель IdTask: Relationship – Связь с классом задание
Методы:	DoTask() – выполнение заданий ExecuteTest() – выполнение теста ViewMark() – просмотр оценок ShowModel() – просмотр моделей Registration() - регистрация GetBetterStudent() – вывод лучших учеников CountStudentfromSchool() – подсчет учеников из одной школы AvgMark() – средний бал

Продолжение таблицы 15

Характеристики	Описание
Связь:	Teacher – связь Ассоциация. Ассоциация показывает, что объекты класса Student связаны с объектами класса Teacher таким образом, что можно перемещаться от объектов одного класса к другому. Task – связь Агрегация. Класс Student включает несколько заданий. При добавлении задания, происходит его привязка к ученику.

В классе Student хранятся все данные об учениках, такие как ФИО, номер телефона, школа, класс. Ученик может просматривать модели в свободном доступе или выполнять практикум. Так же есть возможность просмотреть успеваемость.

Для демонстрации взаимодействия классов между собой были построены диаграмма последовательности и диаграмма кооперации. Для объекта класса Task будет построена диаграмма состояния.

2.3.2 Диаграмма состояний для серверной части мобильного приложения «BioCell»

Диаграмма состояний демонстрирует, как объект переходит из одного состояния в другое. В данной диаграмме отражается процесс смены состояний объекта при наступлении некоторого события и выполнении определенных условий.

Диаграмма состояний класса «Задание» представлена на рисунке 17.

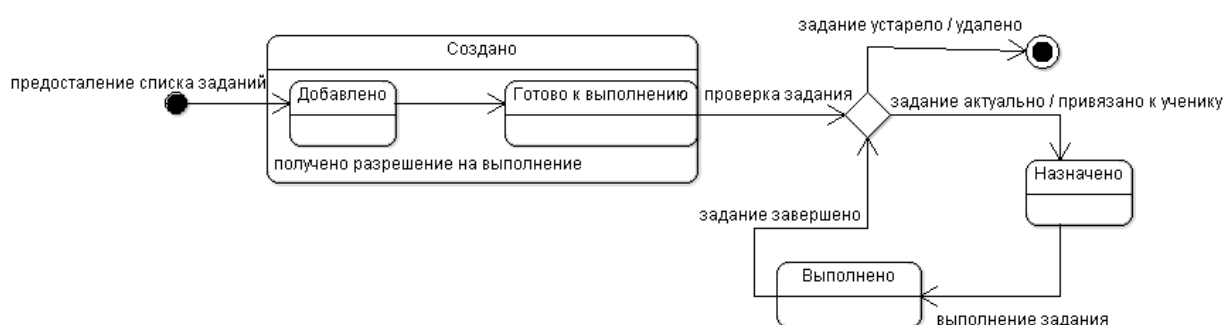


Рисунок 17 – Диаграмма состояний класса «Задание»

Диаграмма описывает состояния, в которых бывает класс Task на протяжении его выполнения. Ее начало совпадает с предоставлением учителем списка заданий внесённых для добавления на сайт. Конец

совпадает с удалением задания из базы данных. Состояние «Создано» обозначает создание тестового задания. Данное состояние включает в себя состояния «Добавлено» и «Готово к выполнению», обозначающих соответственно добавление тестового задания в приложение и готовность тестового задания к выполнению.

Затем идет проверка на актуальность задания. Если задание устарело, то оно удаляется. Если же еще актуально, задание остается в базе и распределяется ученикам. Состояние «Назначено» обозначает, что задание привязано к ученику. Состояние «Выполнено» обозначает, что задание завершено.

После того, как учитель предоставит список заданий, программист добавляет их в приложение и дает разрешения на их выполнение. Перед распределением ученику все задания в списке проверяются на актуальность. Если задание устарело, то оно удаляется. Если актуально, то назначается ученику, после чего выполняется.

2.3.3 Диаграмма деятельности для серверной части мобильного приложения «BioCell»

Диаграмма деятельности показывает разбиение некоторой деятельности на ее составные части. Также на данной диаграмме отображаются логика и последовательность действий. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой простой операции, а переход в следующее состояние срабатывает только при завершении этой операции в предыдущем состоянии [8].

Для демонстрации сценария работы с проектируемыми программными средствами была реализована диаграмма деятельности.

Диаграмма описывает прохождение итогового теста в приложении «BioCell». Она состоит из трех дорожек, которые представляют собой: учителя, программиста и ученика. Начало диаграммы – назначение темы и вида тестирования учителем, конец – тот момент, когда итоговый тест

завершен и баллы получены учеником. Диаграмма деятельности представлена на рисунке 18.

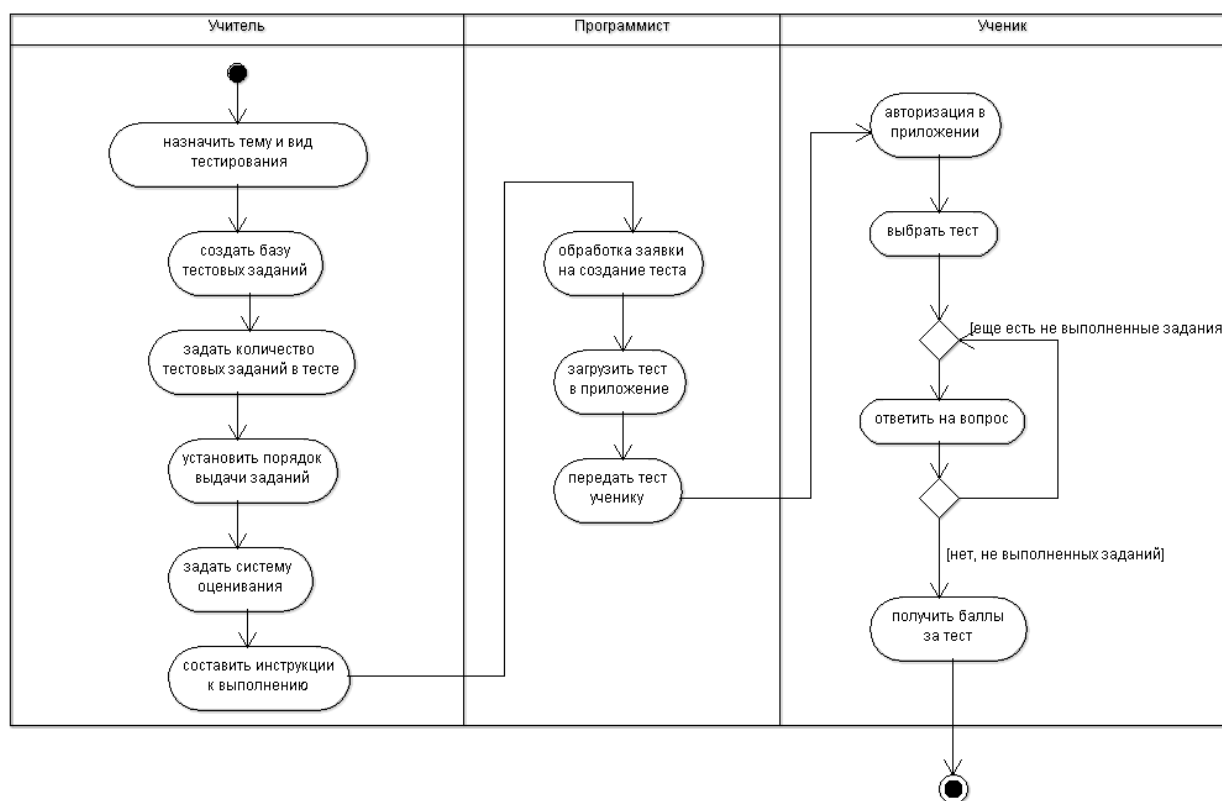


Рисунок 18 – Диаграмма деятельности Прохождение итогового теста

Учитель назначает тему и вид тестирования, проводит анализ учебной дисциплины, после чего систематизирует пройденный материал. Разрабатывает базу тестовых заданий, определяет необходимое количество тестовых заданий, и порядок выдачи заданий. Для теста учитель разрабатывает систему оценивания и инструкции к выполнению. Далее, программист обрабатывает заявку, загружает тест в приложение и передает его ученикам. Ученик после авторизации в приложении, может выбрать тест и начать его выполнение. Если не на все вопросы был дан ответ, то необходимо их доделать, если же нет, не выполненных заданий, то ученик завершает прохождение теста. В конце выполнения, ученик проверяет полученные баллы за тест.

2.3.4 Диаграмма последовательности для серверной части мобильного приложения «BioCell»

Диаграмма последовательности описывает сценарии системы. В данном виде диаграмм демонстрируются объекты и их коммуникационный аспект. Другими словами, отражаются взаимодействующие объекты, которые обмениваются между собой некоторой информацией [8].

Каждый объект имеет собственную линию жизни, означающую период времени, в течение которого объект существует в системе и, следовательно, может потенциально участвовать во всех ее взаимодействиях [8]. Также каждый объект может передавать другому объекту фокус управления, что означает активацию деятельности этого объекта. Для отображения прецедента «Прохождение итогового теста» диаграммы вариантов использования была построена диаграмма последовательности, представленная на рисунке 19.

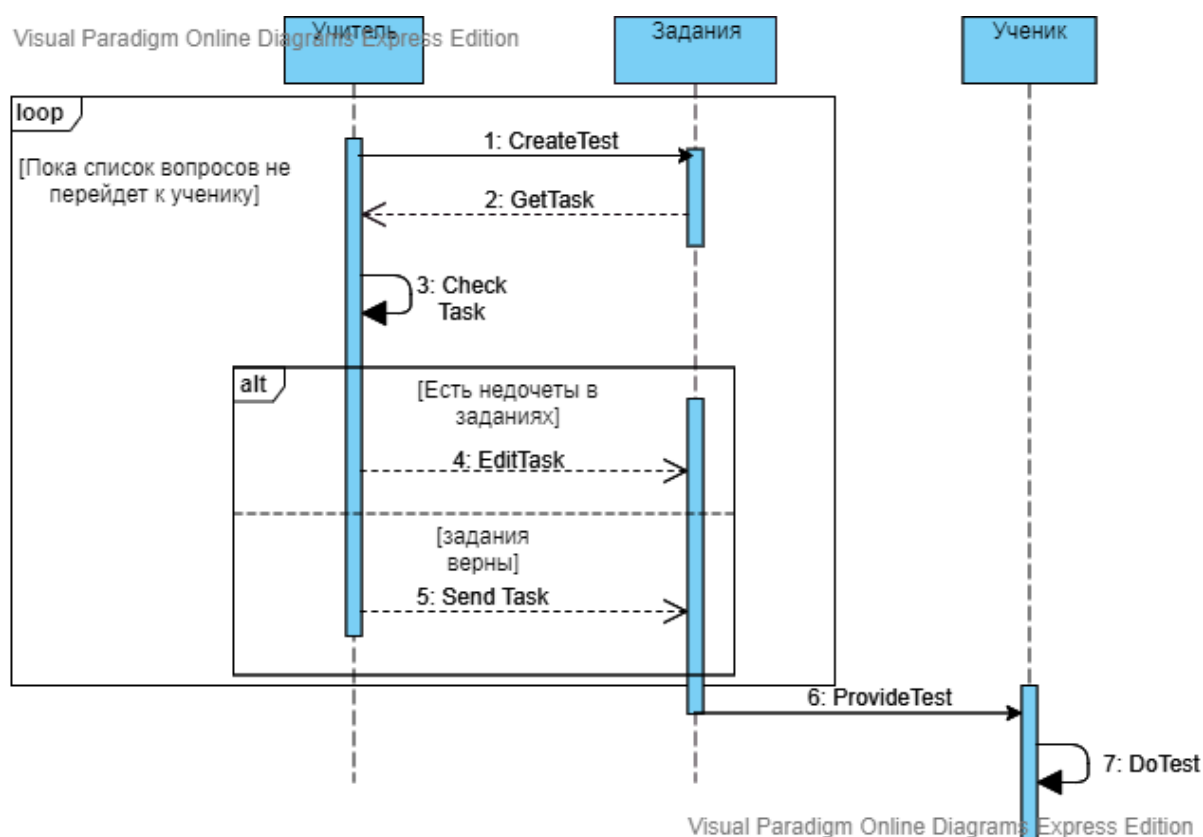


Рисунок 19 – Диаграмма последовательности процесса «Прохождение итогового теста»

Диаграмма описывает процесс прохождения итогового теста. Она включает в себя три объекта: учитель, задания и ученик, которые взаимодействуют в рамках сценария и обмениваются сообщениями.

После появления необходимости в прохождении теста, учитель создает заявку на создание теста и передает её классу задания. Класс задания отправляет предварительный список выбранных вопросов и заданий для теста учителю, который проверяет правильность поставленных заданий и, если есть недочеты в заданиях – отправляет в ответ правки списка заданий, иначе – подтверждение данных. Задания проверяется до того момента, пока тест не будет соответствовать требованиям учителя, после чего он выдается ученику для выполнения.

2.3.5 Диаграмма кооперации для мобильного приложения «BioCell»

Диаграмма кооперации описывает последовательность взаимодействия объектов и их структурные аспекты отношений. Объекты представлены в виде прямоугольников. Динамические связи изображаются в виде линий, где стрелками указывается направление потоков сообщений между объектами, а порядковый номер устанавливает очередность взаимодействия.

Для отображения варианта использования «Выполнение индивидуальных заданий» диаграммы вариантов использования была построена диаграмма кооперации.

Диаграмма описывает процесс выполнения индивидуальных заданий в приложении. Она включает в себя три объекта: ученик, учитель и задание, которые обмениваются сообщениями. Диаграмма кооперации представлена на рисунке 20.

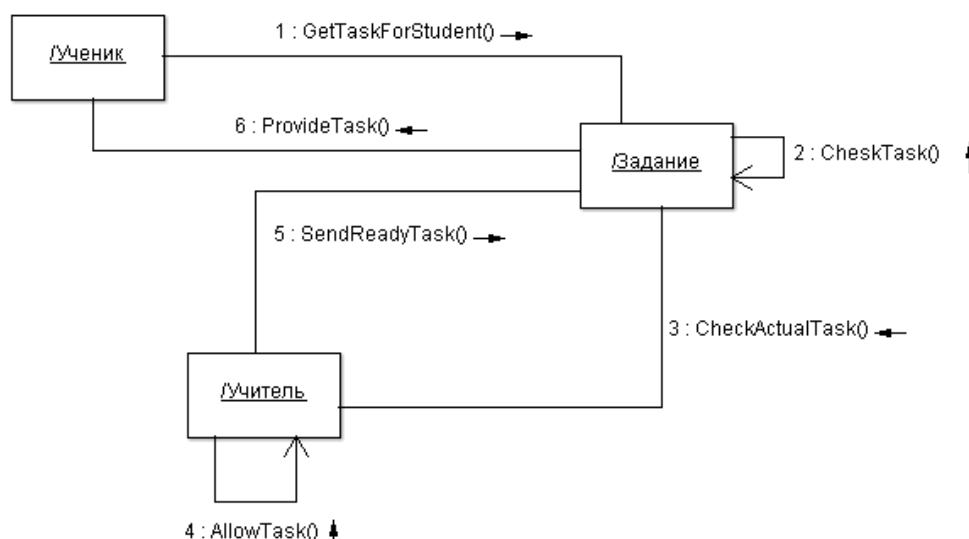


Рисунок 20 – Диаграмма кооперации процесса выполнения индивидуальных заданий

После того, как появляется необходимость в выполнении заданий, ученик создает заявку на выполнение заданий и отправляет ее. Система проверяет список доступных вопросов и отправляет их учителю на проверку актуальности вопросов. Учитель оформляет разрешение на выполнение заданий и отправляет список готовых к выполнению заданий. После чего происходит распределение заданий ученику.

2.4 Проектирование пользовательского интерфейса

На основе диаграммы вариантов использования, было проведено проектирование графического интерфейса пользователя мобильного приложения с дополненной реальностью для изучения структуры растительных клеток и сайта для добавления тестов.

После прохождения авторизации и выдачи всех необходимых прав, пользователю открывается активность для выбора клетки. Для начала работы с 3D объектом пользователю необходимо выбрать модель клетки и нажать на соответствующую кнопку. Интерфейс активности по выбору клетки представлен на рисунке 21.

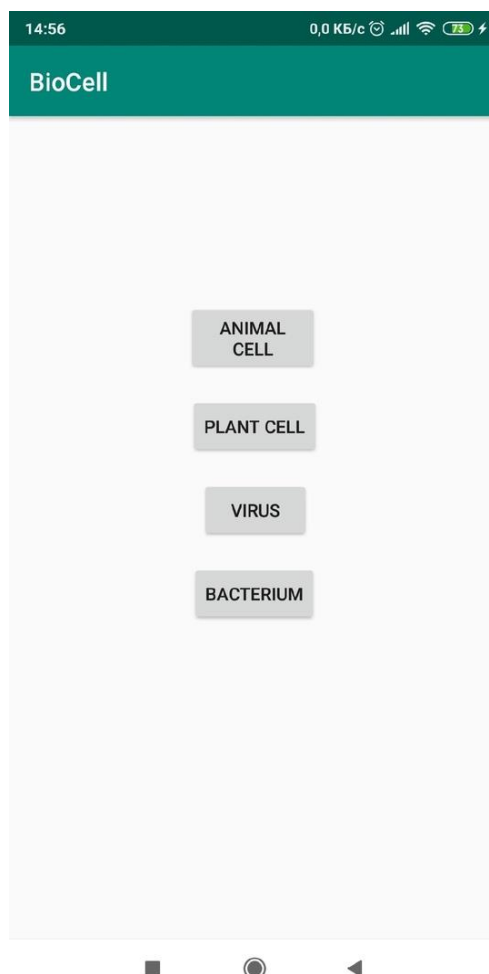


Рисунок 21 – Интерфейс клиентской части с выбором клетки

Для работы с объектами приложения, выполнения практических заданий, а так же для наблюдения процессов, протекающих в клетке и органеллах была спроектирована AR активность. На ней изображены кнопки, по нажатию на которые происходит выбранная реакция. Кроме того изображена сама 3D модель выбранной клетки. На активности можно размещать несколько моделей, так же можно увеличить масштаб модели или перемещать ее в пределах экрана. После нажатия на органеллу, сверху в центре активности, выводится название органеллы и краткая информация о ней. Активность с расположенной на ней клеткой представлена на рисунке 22.

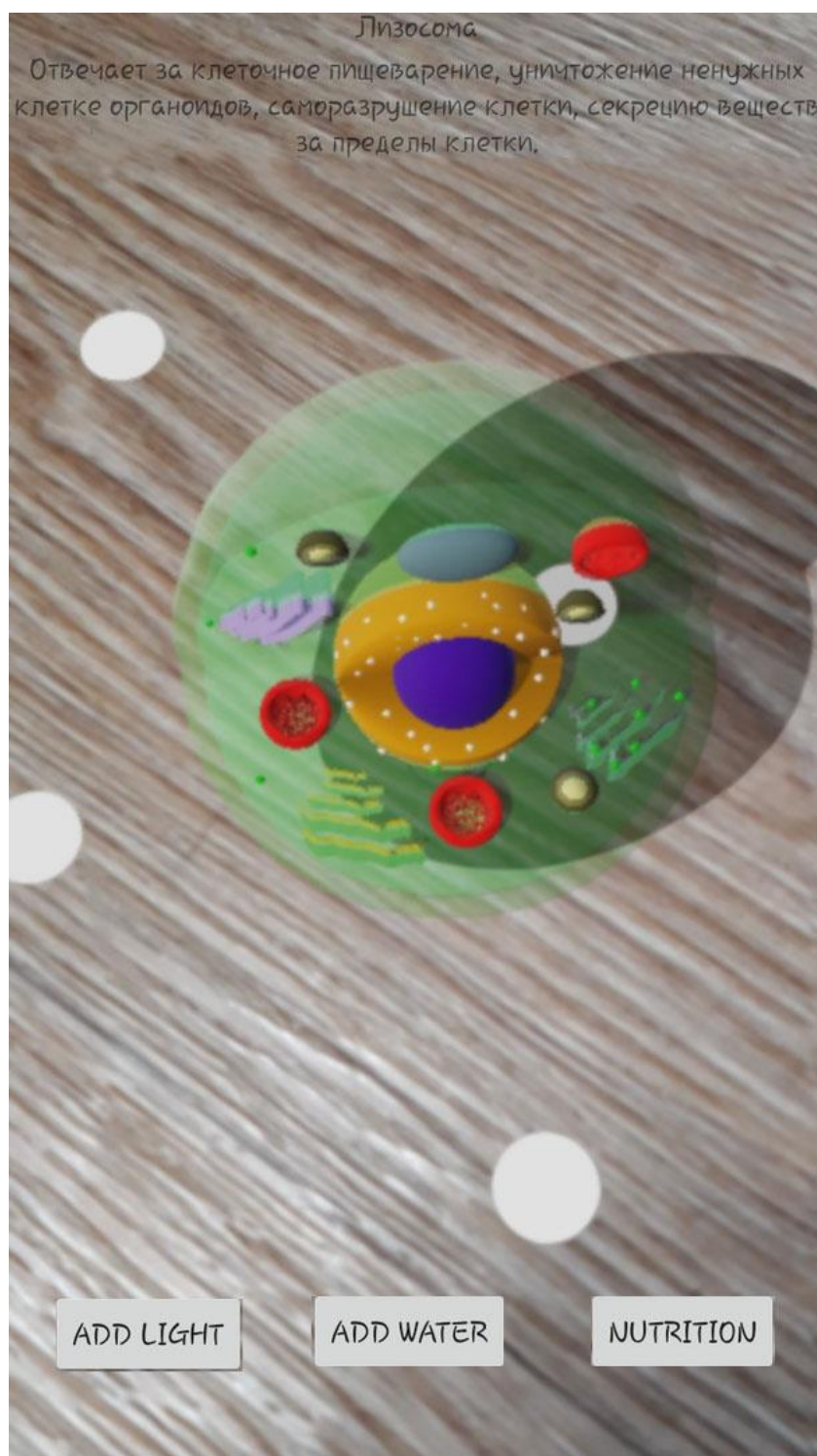


Рисунок 22 – Активность с расположенной на ней клеткой

Образовательная платформа Experiment имеет свой сайт, на котором существует возможность создавать тесты преподавателями. Тестирование используется для оперативной проверки качества знаний учащихся с возможностью машинного ввода данных (ответов) и автоматизированной обработки результата с заранее заданными параметрами качества.

К разработанному в ходе ВКР приложению так же можно составить тесты. Для учителя спроектирован удобный интерфейс, позволяющий добавлять и изменять тесты, вопросы к ним, а так же просматривать результаты проведенных работ. Страница, предоставляющая возможность создания теста представлена на рисунке 23.

Создать тест

* Название:

* Видимость:

* Видимость неправильных ответов:

Предмет:

Год обучения:

Основная картинка:

Краткое описание:

* Попыток разрешено:

* Лимит времени (мин):

☐ Случайный порядок вопросов и ответов

☐ Выставлять оценку в журнале

Рисунок 23 – Страница для создания теста

После создания теста можно добавлять и изменять вопросы к нему. При добавлении вопроса можно устанавливать тип ответа, а так же выбрать количество баллов, которые будут получены учеником в случае

правильного ответа. Страница для добавления вопросов к тесту изображена на рисунке 24.

Добавить вопрос X

* Заголовок

Выбор правильной органеллы.

Задание

Клеточная органелла, предназначенная для выведения вещества из клетки

📎 Прикрепить файл

* Количество баллов

3

* Тип

Вариант BioCell ▾

* Клетка

Растительная ▾

* Органелла

Аппарат Гольджи ▾

Ок

Рисунок 24 – Страница для добавления вопросов к тесту

Кроме того, на сайте добавлена возможность следить за результатами выполнения тестов. Учитель выбирает класс, вид поиска, дату и название теста. После нажатия на кнопку поиска будет получен список учеников, который удовлетворяют выбранным критериям, указан максимальный балл, который можно получить за тест, и количество баллов полученных учеником, что позволяет учителю оценить знания ученика и проставить необходимую оценку. Страница, предоставляющая возможность проверки теста представлена на рисунке 25.

Мои тесты
Библиотека тестов
Выдать тест
Просмотр результатов

Класс: 7Б
Вид поиска: По классу
По ученику
Дата теста: Thu Apr 23 2020
Название теста: Итоговый тест ...
Поиск

☐	ФИО	Максимальный балл	Полученный балл	Дата завершения	Действия
☐	Иванов Иван Иванович	100	95	30.04.20/15:00	Просмотр

Рисунок 25 – Страница проверки тестов

Тестовая технология является быстрым и надежным способом проверки уровня и степени подготовки учащихся. Наличие банка проверочных заданий и возможность проведения автоматизированного тестирования упрощает работу как учителям, так и ученикам. Это позволяет учителям следить за успеваемостью учеников, и в случае отставания в усвоенном материале, повторить тест для ученика еще раз.

3 Реализация программного обеспечения

3.1 Реализация структуры данных

После построения физической модели базы данных в Erwin Data Modeler, необходимо создать саму БД. Для ее построения используем программное обеспечение DBeaver, которое работает с базой данных PostgreSQL. Скрипт созданной БД представлен в приложении Б. Результат создания БД в PostgreSQL можно увидеть на сгенерированной схеме, которая представлена на рисунке 26.

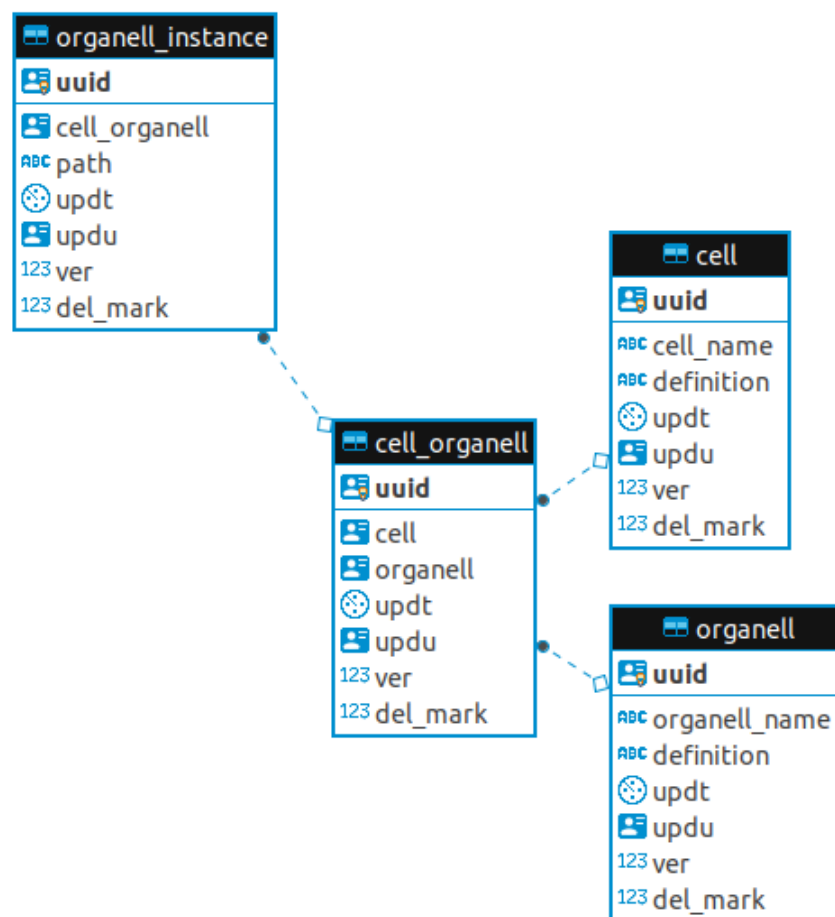


Рисунок 26 – База данных приложения «BioCell»

На платформе Experiment для раздела тестов, вопросов и проверки заданий, создана отдельная база данных. Схема базы данных для раздела «Тесты» представлена на рисунке 27.

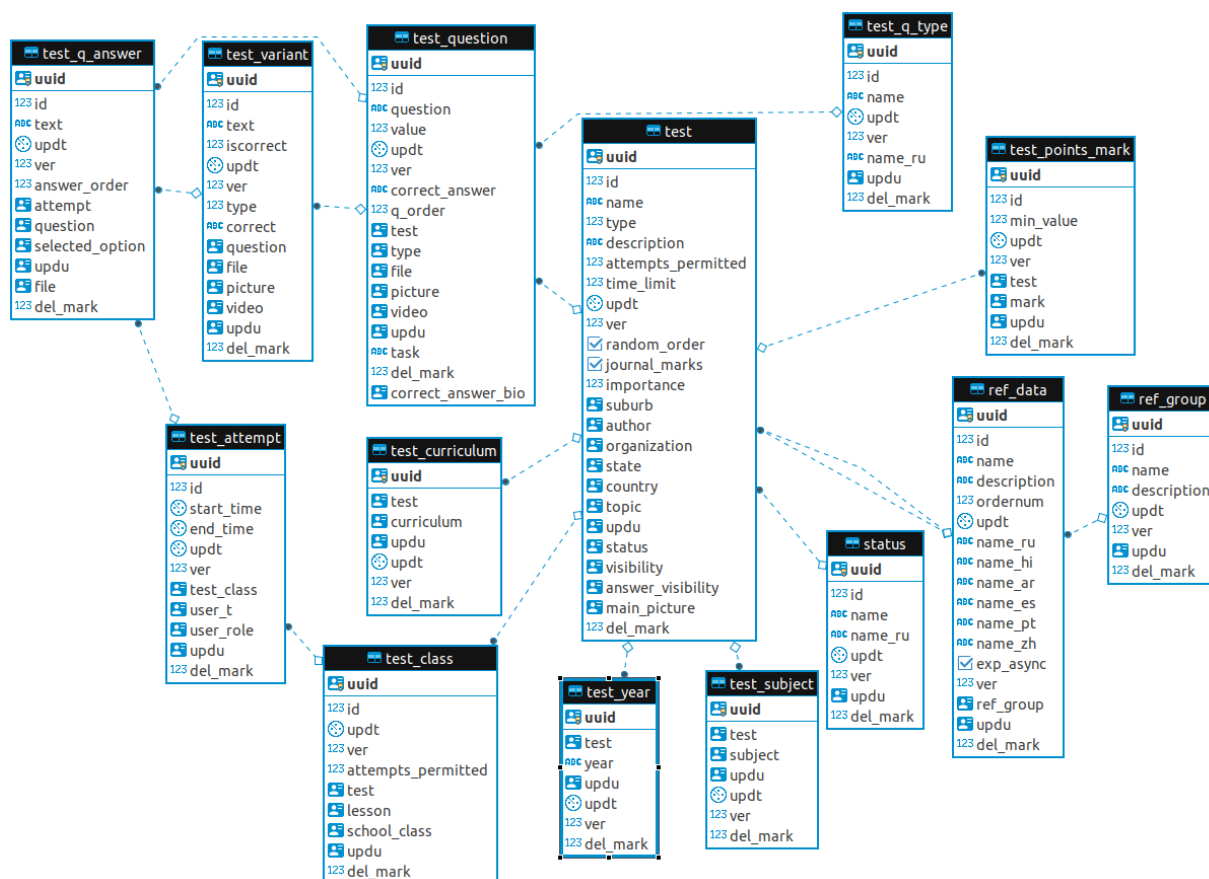


Рисунок 27 – База данных раздела «Тесты»

Заполнение базы данных представлено на примере таблиц test_question для базы данных «Test» и cellorganell для базы данных приложения «BioCell». Заполнение таблицы test_question показано на рисунке 28.

	123 id	123 question	123 value	123 updt	123 ver	123 correct_answer	123 q_order	123 uuid	123 test
950	1 024	Энергитическая станция клетки	2	2020-02-15 17:42:57	0	Митохондрия	1	f3ab3ad0-adde-4636-8ec3-d2cd8b849e6f	c016
951	1 025	Что Защищает клетку от внешних воздейс	1	2020-02-15 17:42:57	0	Мембрана	2	47430f70-3d28-4fed-a082-70353498feed	c016
952	1 026	Переносит и преобразует белки, произво	1	2020-02-15 17:42:57	0	Аппарат Гольджи	3	74a4e513-3db0-4ec3-aac4-25ead71d1a4b	c016
953	1 027	Содержит зеленый пигмент, называемый	3	2020-02-15 17:42:57	0	Хлоропласт	4	1c862b61-06f0-4353-b017-944f4f468f1b	c016

Рисунок 28 – Заполнение таблицы test_question

В таблице отображен задание, тип задания, тип ответа и ответ к нему. Кроме того указаны данные о том кто и когда добавил задание.

Заполнение таблицы cellorganell показано на рисунке 29.

	uuid	organelle_name	organelle_definition	updt
1	5f9c3a32-59fd-496c-8f3a-d1059b5bb4ed	Мембрана	Защищает клетку от внешних воздействий	2020-03-18 16:17:36
2	d4a635f4-e7c2-4b4a-8a56-90b7dad73d87	Ядро	Наиболее важная часть клетки, в ней находятся хромосомы, которые несут в себе наследственную информацию	2020-03-18 16:17:35
3	502fbc6d-4cdc-49d4-8bf7-cad01f5def33	Ядрышко	Образует рибосомные субъединицы	2020-03-18 16:17:35
4	009dd39b-9fd2-431d-871b-20511a6d3f74	Центриоль	Образуют клеточный центр, который нужен для деления клетки	2020-03-18 16:17:35
5	062a400e-70f3-4349-9205-040f30f63a7a	Цитоплазма	Объединяет все органеллы и обеспечивает их работу	2020-03-18 16:17:36
6	9317848f-703d-49b3-b2d4-3d826035440b	Лизосома	Отвечает за клеточное пищеварение, утилизацию отходов	2020-03-18 16:17:36
7	414d32dd-a383-4501-baa5-9a4b1f13ef1a	Шероховатая эндоплазматическая сеть	Отвечает за синтез белков, так как на ее поверхности находятся рибосомы	2020-03-18 16:17:36
8	ee169a1e-1ea3-4dbf-83db-692505738bf9	Гладкая эндоплазматическая сеть	Отвечает за синтез жиров (липидов) и углеводов	2020-03-18 16:17:36
9	4d8082b1-afe6-417e-a726-d686a55b40fc	Вакуоль	Отвечает за хранение питательных веществ и удаление отходов	2020-03-18 16:17:35
10	3eb9cab9-938d-46fb-9b09-6ea81d3751aa	Аппарат Гольджи	Переносит и преобразует белки, производимые в клетке	2020-03-18 16:17:35
11	4c9e32ce-3659-4203-8f22-ccc82893b4ce	Хлоропласт	Содержит зеленый пигмент, называемый хлорофиллом, который участвует в фотосинтезе	2020-03-18 16:17:36
12	428917b8-2e4f-45fb-a7d4-cb87ebee424b	Митохондрия	Энергетическая станция клетки	2020-03-18 15:15:40

Рисунок 29 – Заполнение таблицы cellorganell

В таблице указан автоматический генерируемый ID для органелл, название органеллы и ее описание. Так же указано кто и когда дополнял базу, и версия обновления.

Для проверки работоспособности сервисов приложения «BioCell», обратимся к ним через сервер и получим JSON объект. Вызов процедуры Organell представлен на рисунке 30.

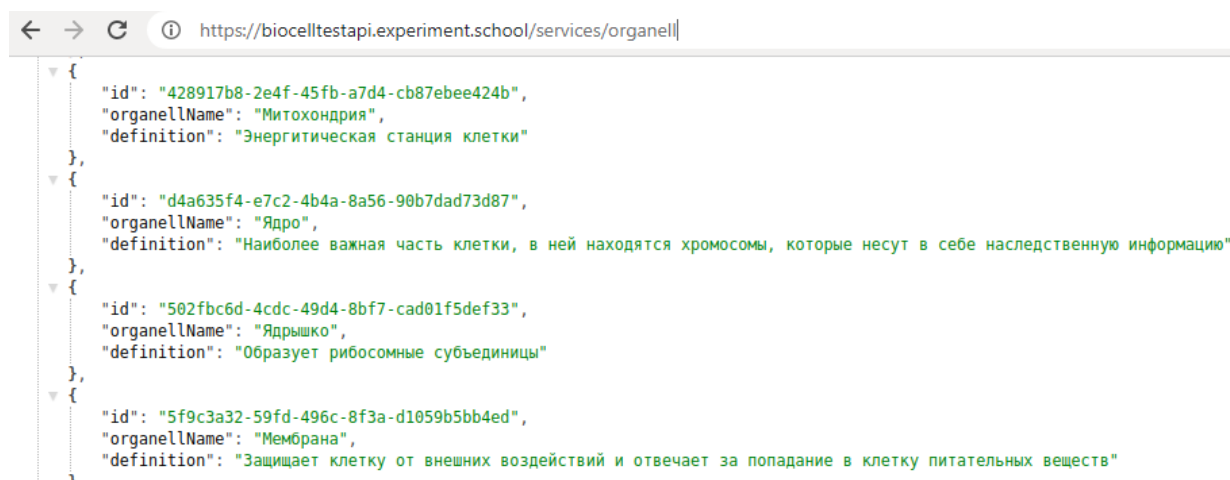


Рисунок 30 – Вызов процедуры Organell

При вызове процедуры Organell, выводятся весь список органелл, хранящийся в базе данных и информация о них: ID органеллы, её название и описание.

Вызов процедуры getForCell представлен на рисунке 31.

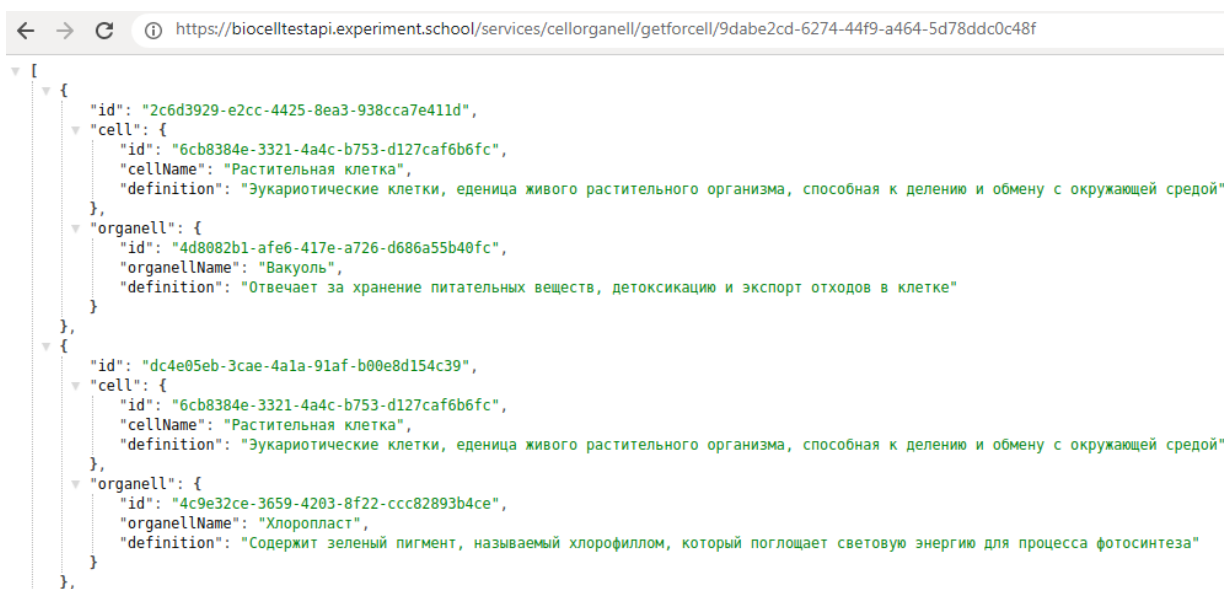


Рисунок 31 – Вызов процедуры getForCell

При использовании процедуры getForCell и передаче в нее ID клетки, выводится вся информация о конкретной клетке, ее органеллах и функциях. Процедура помогает узнать, какие именно органеллы находятся в этой клетке, и для чего они предназначены.

Вызов процедуры cellorganell представлен на рисунке 32.

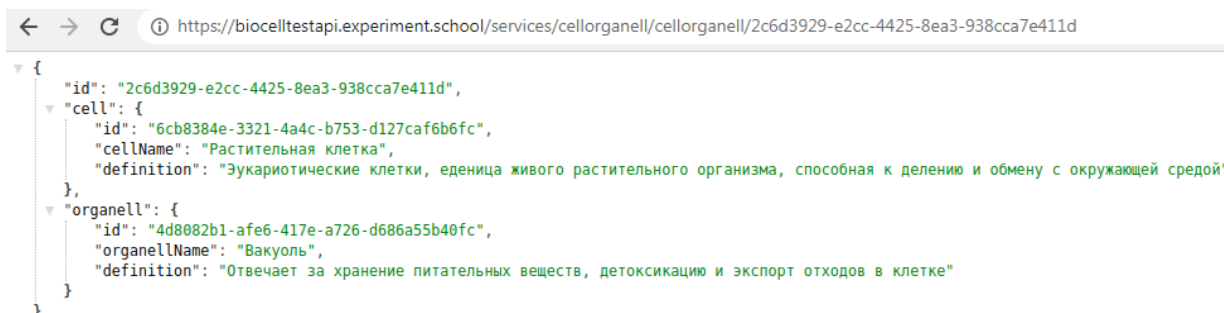


Рисунок 32 – Вызов процедуры cellorganell

Таблица cellorganell показывает все связи клеток и органелл. При вызове команды cellorganell и передаче в нее ID связи, выводится вся информация о клетке и связанной с ней органеллой.

Результаты работы реализованных процедур соответствует ожидаемым результатам, что показывает правильность их выполнения.

3.2 Реализация пользовательского интерфейса

Нами были разработаны все модели клеток и органелл. Рассмотрим разработку 3D модели на примере растительной клетки. Пример клетки с наложенными на нее материалами представлен на рисунке 33.

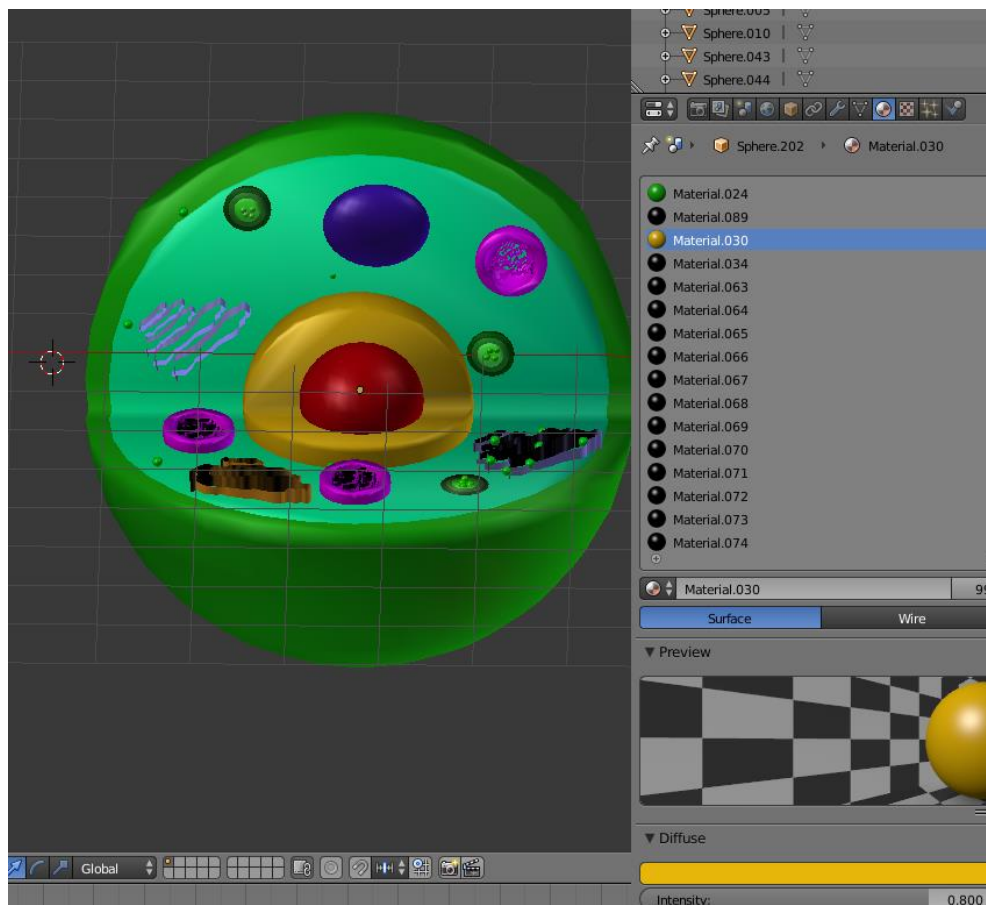
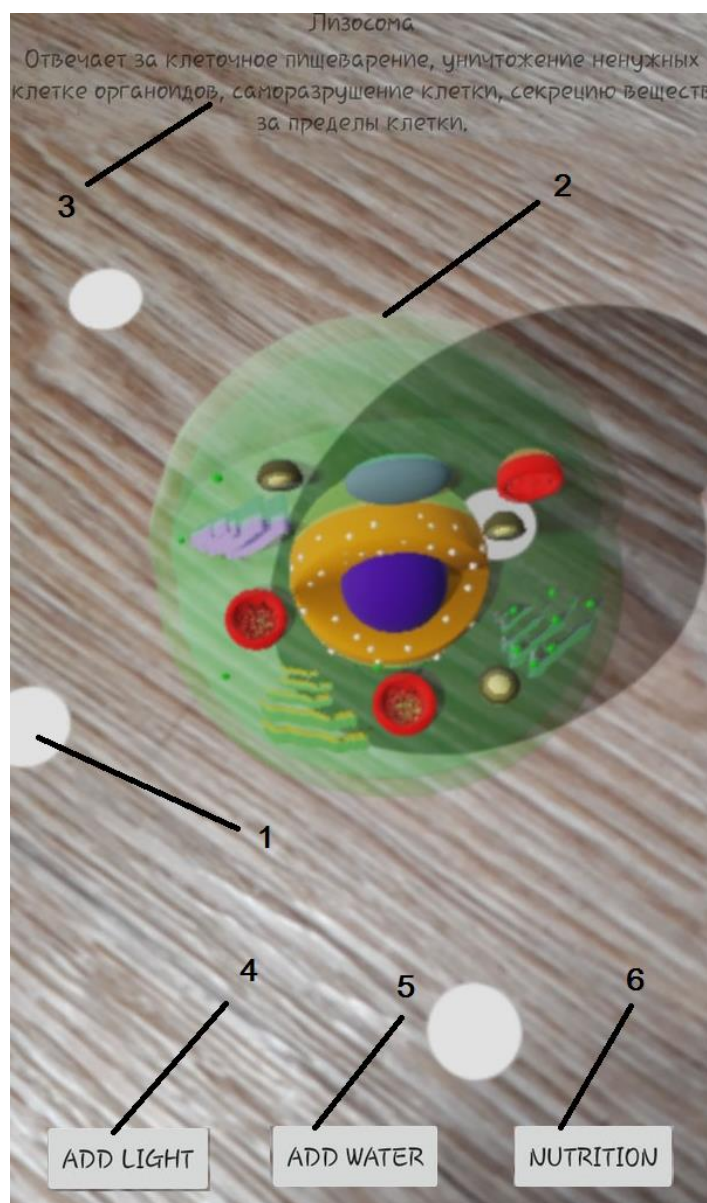


Рисунок 33 – Пример разработки 3D модели

За основу модели взята сфера, где было уменьшено количество полигонов, вершин и граней для увеличения скорости отрисовки объектов и уменьшения требований к устройству. Для того чтобы органеллы выглядели как реальные, были разработаны материалы и наложены на объекты. Для получения полноценного объекта клетки были объединены все модели органелл между собой.

После входа в приложения и разрешения всех прав первым экраном, который увидит пользователь, будет активность с выбором клетки. Для начало работы с объектами пользователь должен выбрать нужную ему

клетку и нажать необходимую кнопку. После выбора клетки появляется AR активность с изображенной 3D моделью, для ее отображения необходимо найти ровную поверхность и нажать на экран. Так же для возможности просмотра процессов, протекающих в клетке, были добавлены необходимые кнопки. Интерфейс AR активности представлен на рисунке 34.



- 1 – AR поверхность для отображения на ней объектов;
- 2 – 3D модель растительной клетки;
- 3 – подсказка, дающая информацию о предназначении органеллы;
- 4 – кнопка для добавления света, и просмотра реакции клетки;
- 5 – кнопка для добавления воды, и просмотра реакции клетки;
- 6 – кнопка для добавления питательных веществ, и просмотра реакции клетки.

Рисунок 34 – AR активность с расположенной на ней клеткой

На сайте для добавления и редактирования тестов, и тестовых заданий, и для просмотра результатов сделаны web-страницы. Интерфейс страницы сайта рассмотрим на примере добавления вопроса к тесту. Страница для добавления вопросов к тесту изображена на рисунке 35.

1 — Добавить вопрос

2 — * Заголовок

3 — Выбор правильной органеллы.

4 — * Задание

5 — Клеточная органелла, предназначенная для выведения вещества из клетки

6 — Прикрепить файл

7 — * Количество баллов

8 — 3

9 — * Тип

10 — Вариант BioCell

* Клетка

Растительная

* Органелла

Аппарат Гольджи

10 — Ок

- 1 — заголовок страницы, дающий название страницы;
- 2 — текстовые объекты, дающие названия для объектов;
- 3 — раздел для описания вопроса;
- 4 — раздел для подробного описания задания;
- 5 — кнопка для прикрепления файла, необходимого для задания;
- 6 — раздел для установления сложности задания;
- 7 — раздела для выбора типа вопроса;
- 8 — раздел для выбора клетки, для которой предназначен вопрос;
- 9 — раздел для выбора органелл, для которых предназначен вопрос;
- 10 — кнопка для подтверждения и сохранения вопроса.

Рисунок 35 — Пример добавления вопроса к тесту

Спроектирован удобный интерфейс для пользователей данного приложения. Для добавления теста и вопросов необходимо заполнить все поля помеченные звездочкой.

3.3 Реализация программных средств

3.3.1 Диаграмма компонентов мобильного приложения для изучения структуры растительных клеток

Диаграмма компонентов описывает особенности физического представления системы. Архитектура разрабатываемой системы, устанавливает зависимости между программными компонентами, в роли которых выступают элементы интерфейса и основные классы программы. Модуль или компонент соответствуют файлам. Компонент – это физически существующая часть системы, благодаря которой обеспечивается реализация классов и отношений, а также достигается функциональность моделируемой информационной системы.

Программные средства разрабатываемого мобильного приложения с дополненной реальностью для изучения структуры растительных клеток состоят из компонентов, которые представлены на рисунке 36.

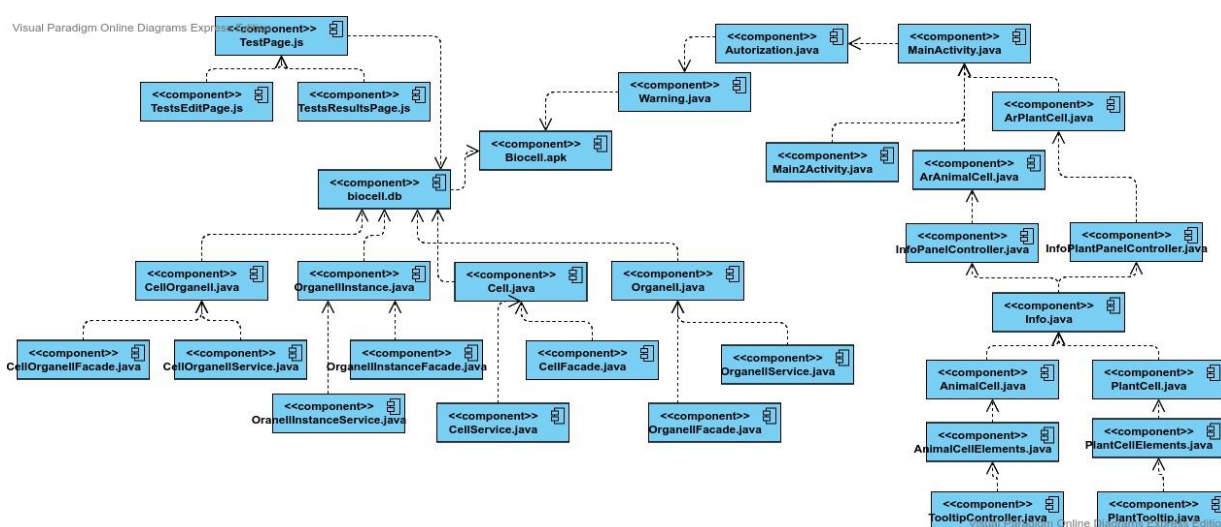


Рисунок 36 – Диаграмма компонентов приложения «BioCell»

В данной выпускной квалификационной работе рассматривается разработка серверной части мобильного приложения с дополненной

реальностью для изучения структуры растительных клеток. Диаграмма компонентов для серверной части представлена на рисунке 37.

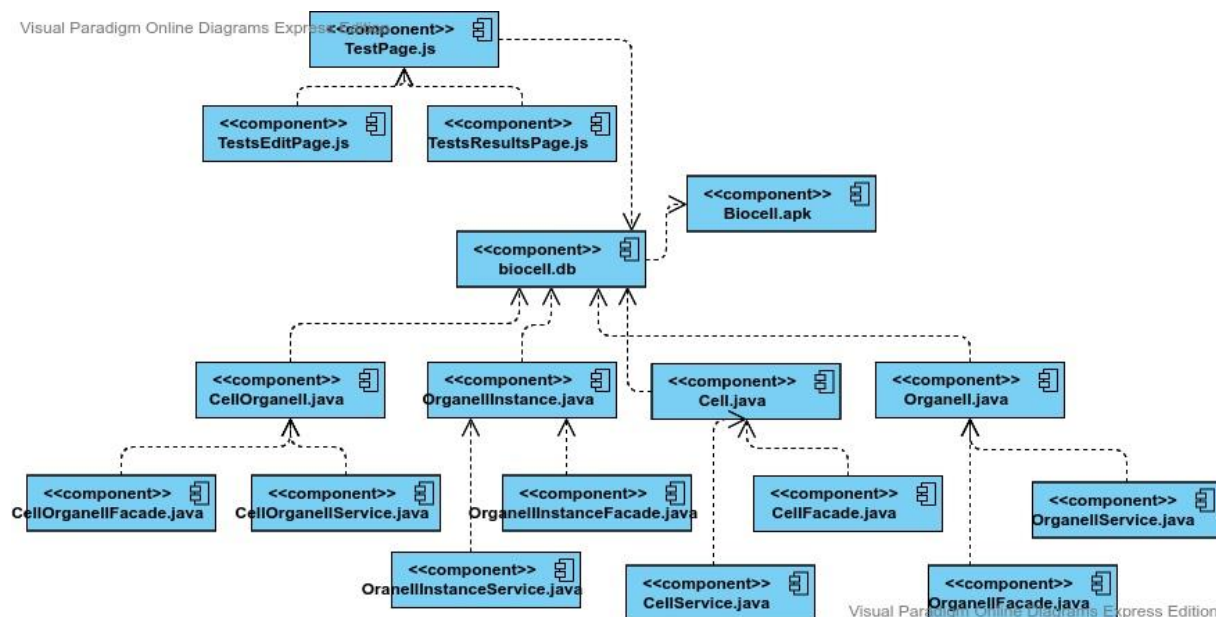


Рисунок 37 – Диаграмма компонентов серверной части приложения «BioCell»

Описание компонентов представлено ниже:

- BioCell.apk – основной исполняемый файл;
- TestPage.js – содержит структуру и описание страницы для тестов на сайте;
- TestsEditPage.js – содержит структуру и описание страницы для создания и редактирования тестов на сайте;
- TestsResultPage.js – содержит структуру и описание страницы для проверки результатов пройденных тестов;
- Biocell.db – содержит базу данных для приложения «BioCell», с описанием клеток и органелл
- CellOrganell.java – содержит класс для связи клеток и органелл;
- OrganellInstance.java – содержит класс для подгрузки объектов органелл с сервера;
- Cell.java – содержит класс для создания объектов клеток;
- Organell.java – содержит класс для создания объектов органелл;

- CellOrganelFacade.java – содержит процедуры для класса CellOrganel;
- CellOrganelService.java – содержит класс для связи файла CellOrganelFacade.java с сервером;
- CellFacade.java – содержит процедуры для класса Cell;
- CellService.java – содержит класс для связи файла CellFacade.java с сервером;
- OrganelFacade.java – содержит процедуры для класса Organel;
- OrganelService.java – содержит класс для связи файла OrganelFacade.java с сервером;
- OrganelInstanceFacade.java – содержит процедуры для класса OrganelInstance;
- OrganelInstanceService.java – содержит класс для связи файла OrganelInstanceFacade.java с сервером.

3.3.2 Диаграмма развертывания для мобильного приложения «BioCell»

Диаграмма развёртывания – содержит графические изображения процессоров, устройств, процессов и связей между ними. Диаграмма развертывания является единой для системы в целом и всецело отражает особенности ее реализации. Диаграмма представляет физическое расположение системы, показывая, на каком физическом оборудовании запускается та или иная составляющая программного обеспечения. Узлы устройств – это физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения. Диаграмма развертывания мобильного приложения с дополненной реальностью для изучения структуры растительных клеток представлена на рисунке 38.

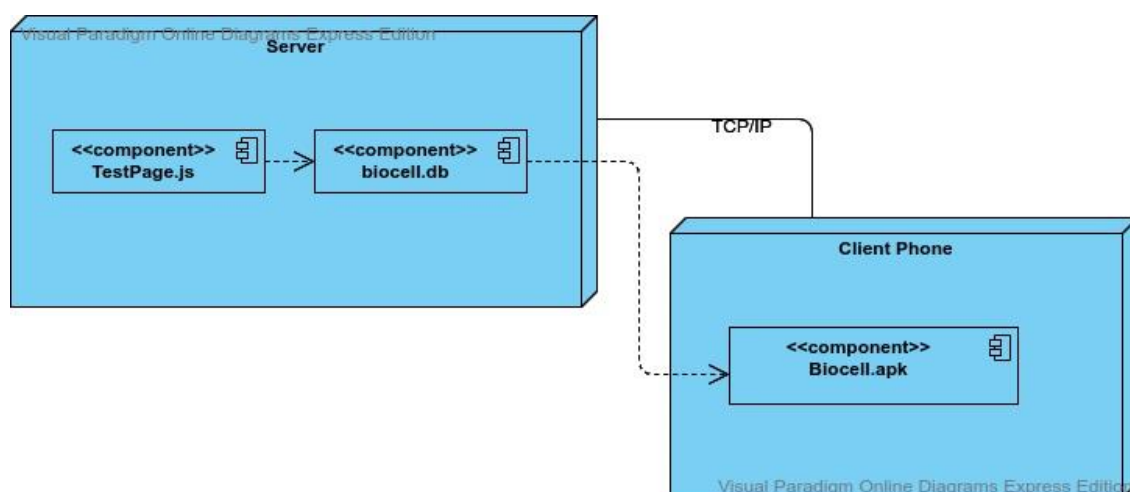


Рисунок 38 – Диаграмма развертывания

Система состоит из мобильного телефона клиента, с помощью которого он осуществляет взаимодействие с ней через приложение «BioCell». Так же в систему включает удалённый сервер, который позволяет принимать запросы пользователя, отправлять запросы в БД, получать ответ и передавать его пользователю. Сервер БД реализован в PostgreSQL принимает запросы сервера, обрабатывает их и возвращать ответ. Взаимодействие мобильного телефона пользователя с сервером происходит через TCP/IP соединение.

4 Тестирование программного обеспечения

Тестирование является способом определения качества программного продукта. Искусство тестирования находится в прямой зависимости от умения программиста разрабатывать эффективные тесты. Суть тестирования программного продукта заключается не в том, чтобы показать, что она работает, а скорее наоборот – тестирование начинается с предположения, что в программном продукте есть ошибки, а затем уже обнаруживается их максимально возможное число. Таким образом, тестирование – это процесс исполнения программы с целью обнаружения ошибок [9].

В данной выпускной квалификационной работе рассмотрены такие методологии тестирования, как структурное и Unit тестирование. Также проведён анализ качества программного кода путём определения различных метрик.

4.1 Структурное тестирование

Структурное тестирование (тестирование по принципу «белого ящика») характеризуется степенью, в которой тесты покрывают логику (исходный текст) программы. Данная методология тестирования предполагает выполнение каждого пути в программе.

В данной выпускной квалификационной работе тестирование по принципу белого ящика включает в себя метод покрытия операторов, метод покрытия условий и метод покрытия маршрутов.

Тестирование может быть проведено на примере прецедента диаграммы вариантов использования «Просмотр модели клеток». Блок-схема алгоритма представлена на рисунке 39.

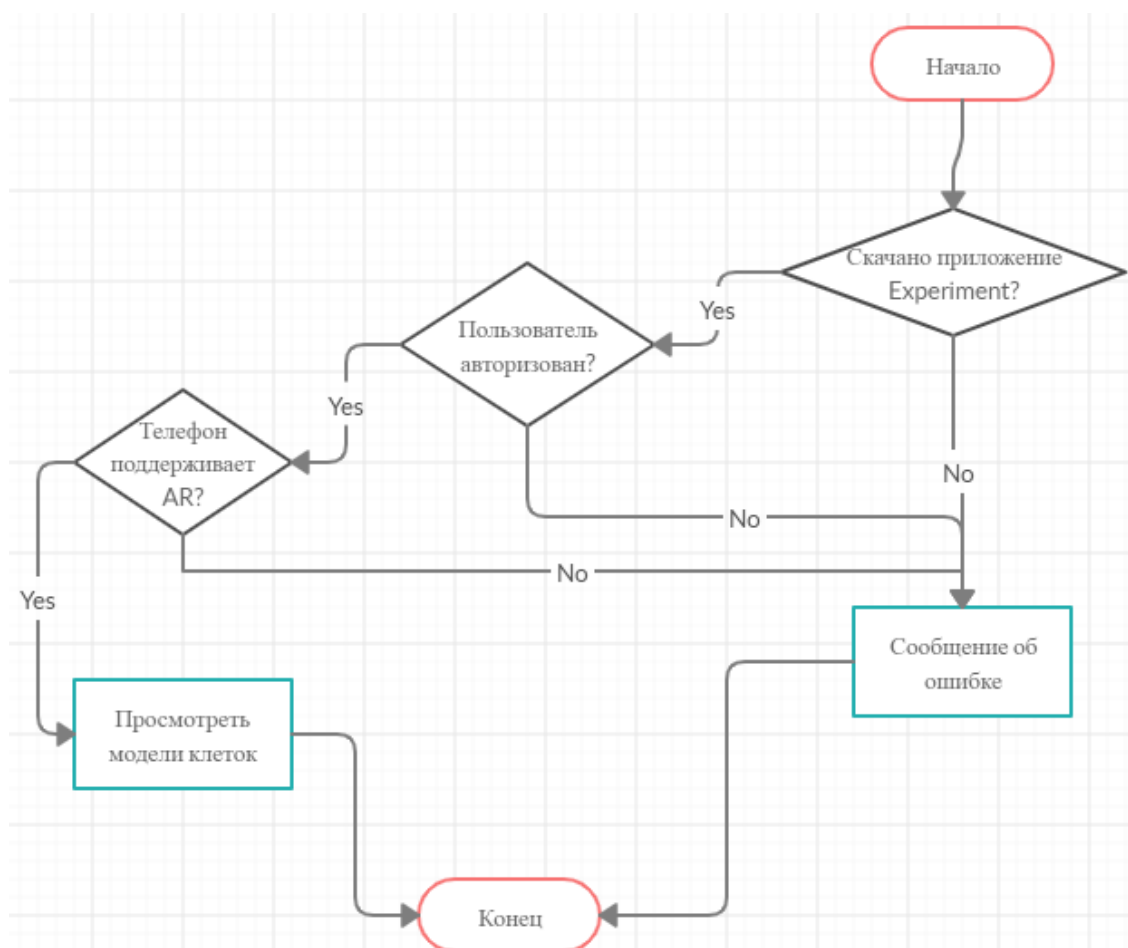


Рисунок 39 – Блок-схема прецедента «Просмотр модели клеток»

На блок-схеме изображено, какие условия проверяются для входа в приложение. Первым условием проверяется, скачано ли приложение Experiment. Если приложения нет, то программа выводит ошибку. В случае успешного выполнения условия, проверяется второе условие, авторизован ли пользователь в нем. Наличие авторизованной учетной записи является обязательной частью, для успешной работы приложения. В конце проверяется, поддерживает ли телефон режим дополненной реальности и выданы ли все необходимые права, для работы с приложением. Если все условия выполнены, то приложение открывает активность с выбором клетки.

Управляющий граф алгоритма, отображающий последовательность операторов, представлен на рисунке 40.

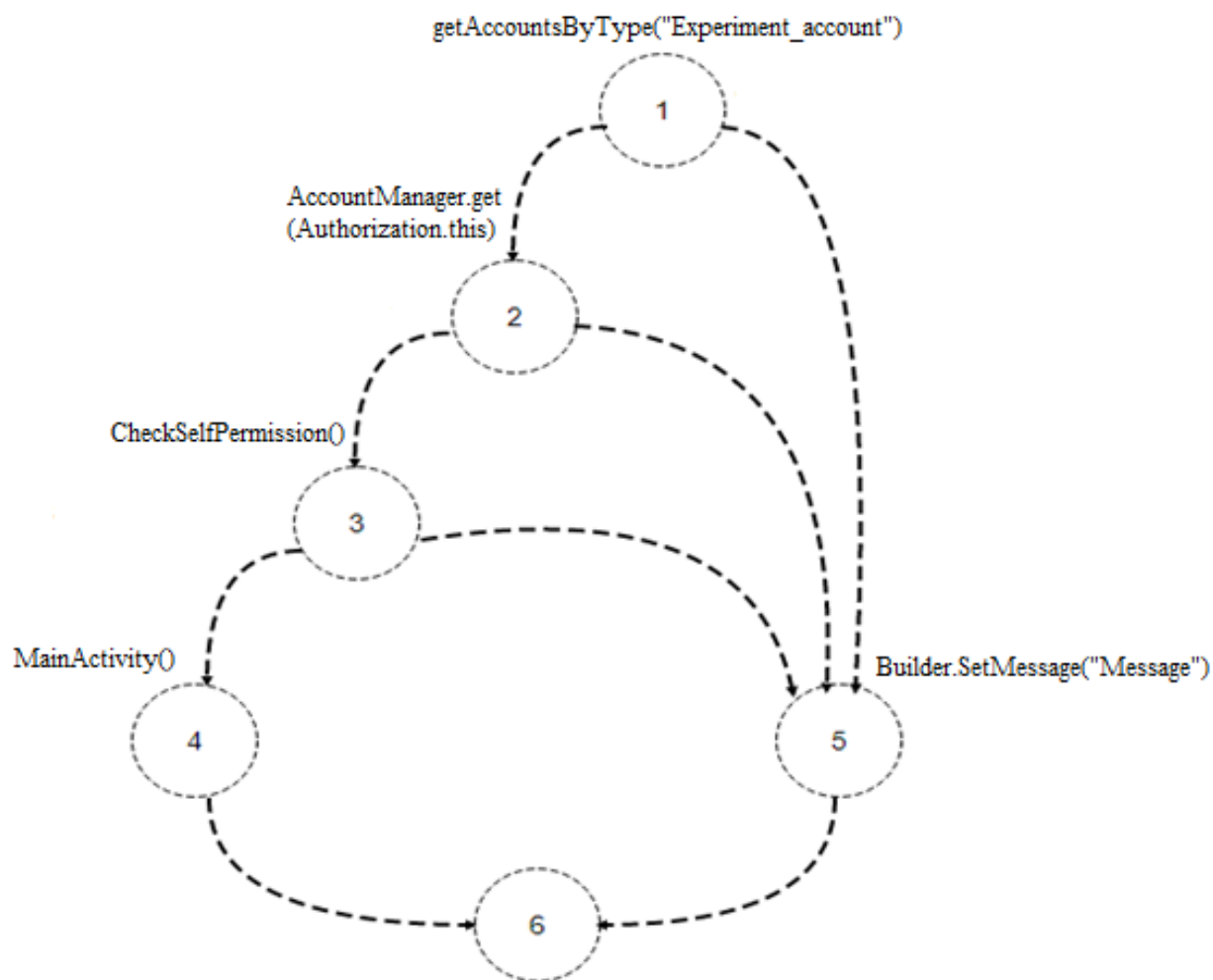


Рисунок 40 – Управляющий граф прецедента «Просмотр модели клеток»

Основываясь на блок-схеме и управляющем графе, было проведено тестирование методом покрытия операторов, методом покрытия условий и методом покрытия маршрутов. Код алгоритма выполнения прецедента «Просмотр модели клеток» представлен в листинге В.13.

4.1.1 Метод покрытия операторов

Метод покрытия операторов подразумевает выполнение каждого оператора программы, по крайней мере, один раз. Для выполнения каждого оператора было составлено два теста. Результаты приведены в таблице 16.

Таблица 16 – Результат тестирования методом покрытия операторов

Операторы	Тест	Фактический результат	Результат тестирования
1, 2, 3, 4	Приложение скачано, Пользователь авторизован, телефон поддерживает режим	Клетка успешна выбрана	Успешно. (рисунок Г.1)
1, 5, 6	Приложение не скачано.	Клетка не была выбрана	Успешно. (рисунок Г.2)

По результатам метода покрытия операторов все тесты были успешно пройдены. Полученные результаты полностью соответствуют ожидаемым.

4.1.2 Метод покрытия условий

Суть метода покрытия условий заключается в том, чтобы составить число тестов, достаточное для того, чтобы все возможные результаты каждого условия в решении выполнялись, по крайней мере, один раз. Также должно соблюдаться условие выполнения каждого оператора, по крайней мере, один раз.

Для выполнения каждого условия было составлено четыре теста. Результаты приведены в таблице 17.

Таблица 17 – Результат тестирования методом покрытия условий

Условия	Тест	Фактический результат	Результат тестирования
1 – нет	Приложение не скачано.	Вывод ошибки приложения.	Успешно. (рисунок Г.2)
1 – да, 2 – нет	Приложение скачано, Пользователь не авторизован.	Вывод ошибки авторизации.	Успешно. (рисунок Г.3)
1 – да, 2 – да, 3 – нет	Приложение скачано, Пользователь авторизован, Телефон не поддерживает режим.	Вывод ошибки в получении прав.	Успешно. (рисунок Г.4)
1 – да, 2 – да, 3 – да	Приложение скачано, Пользователь авторизован, Телефон поддерживает режим.	Клетка успешно выбрана.	Успешно. (рисунок Г.1)

По результатам тестирования по методу покрытия условий, все тесты были успешно пройдены. Если пользователь не авторизован или права не выданы, выводится соответствующее сообщение об ошибке, иначе – доступ разрешен.

4.1.3 Метод покрытия маршрутов

Метод покрытия маршрутов является наиболее полным и результативным. Он предполагает прохождение по каждому возможному пути программы. Маршрутом называют уникальную последовательность ветвей логических условий (branches) от входа в функцию до выхода из неё.

Для выполнения всех маршрутов было составлено четыре теста. Результаты приведены в таблице 18.

Таблица 18 – Результат тестирования методом покрытия условий

Маршрут	Тест	Фактический результат	Результат тестирования
1, 5, 6	Приложение не скачано.	Вывод ошибки приложения.	Успешно. (рисунок Г.2)
1, 2, 5, 6	Приложение скачано, Пользователь не авторизован.	Вывод ошибки авторизации.	Успешно. (рисунок Г.3)
1, 2, 3, 5, 6	Приложение скачано, Пользователь авторизован, Телефон не поддерживает режим.	Вывод ошибки прав.	Успешно. (рисунок Г.4)
1, 2, 3, 4, 6	Приложение скачано, Пользователь авторизован, Телефон поддерживает режим.	Клетка успешно выбрана.	Успешно. (рисунок Г.1)

По результатам тестирования по методу покрытия маршрутов все тесты были успешно пройдены. Полученные результаты полностью соответствуют ожидаемым.

4.2 Качественный анализ кода

Определение качества кода является важной частью в мире программирования. В частности, от сложности кода системы зависит производительность труда разработчиков, а, следовательно, трудоемкость и сроки разработки. Между сложностью и надежностью программной системы также существует тесная связь. Очевидно, что чем выше сложность программной системы, тем труднее обеспечить ее надежность [10].

Сложность – это одна из главных причин ненадежности программного обеспечения. Сложность почти не поддается ни точному определению, ни измерению. Однако можно сказать, что мерой сложности объекта является количество интеллектуальных усилий, необходимых для понимания этого объекта.

Важным правилом программирования является соблюдение простоты кода. Более простой считается та программа, которая легче для понимания, отладки, сопровождения и модификации.

При оценке сложности программ в данной выпускной квалификационной работе выделены два типа метрик:

- 1) метрика размера программ;
- 2) метрика сложности потока управления программ.

Качественный анализ кода был проведен для листинга 4.1, реализующего просмотр модели клеток.

4.2.1 SLOC-метрика

Количество строк кода (англ. Source Lines of Code – SLOC) — это метрика программного обеспечения, используемая для измерения его объема с помощью подсчёта количества строк в тексте исходного кода. Как правило, этот показатель используется для прогноза трудозатрат на разработку конкретной программы на конкретном языке программирования, либо для оценки производительности труда уже после того, как программа

написана. Физическими строками считаются все непустые строки текстового файла.

Применим SLOC-метрику к разрабатываемому программному коду для просмотра моделей клетки (рисунок 39). Результаты данной метрики представлены в таблице 19.

Таблица 19 – SLOC-метрики

Число физических строк	82
Число логических строк	59
Число пустых строк	0
Число строк-комментариев	9

Измеряя логические строки кода, предпринимается попытка посчитать количество собственно операторов в программе, но, разумеется, их определение зависит от конкретного языка программирования.

4.2.2 Метрика Холстеда

В качестве метрики размера программы была рассмотрена метрика Холстеда. В данной метрике за базу принят подсчет количества операторов и операндов, используемых в программе, т.е. определение размера программы.

Основу метрики Холстеда составляют четыре измеряемые характеристики программы: n_1 – число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов); n_2 – число уникальных операндов программы (словарь операндов); N_1 – общее число операторов в программе; N_2 – общее число операндов в программе. Подсчет каждой измеряемой характеристики представлен в таблице 20.

Таблица 20 – Операнды и операторы

Операторы	Число вхождений	Операнды	Число вхождений
=	7	@Override	1
.	62	accountManager	2
()	44	savedInstanceState	1
->	1	Permissions[]	1
[]	8	accounts	3
!=	2	requestCode	1
==	1	grantResults	2
&&	1	builder	4
onCreate()	1	intent	2
NextLevel()	2	dialog	3
onClick()	1	id	1
:	2		
>	1		
If()	4		
else	4		
new	8		
return	1		
$n1 = 17$	$N1 = 150$	$n2 = 11$	$N2 = 21$

На основании этих характеристик рассчитываются следующие оценки.

Длина программы рассчитывается по формуле:

$$N = N_1 + N_2.$$

Словарь программы рассчитывается по формуле:

$$n = n_1 + n_2 .$$

Вычислительная сложность рассчитывается по формуле:

$$HV = N \times \log(2) n \text{ (бит)}.$$

Под битом подразумевается логическая единица информации – символ, оператор, операнд. Объем программы по Холстеду определяет объем кода алгоритма. Размер метрики для одной функции должен находиться в пределах от 20 до 1000.

В результате вычисления измеряемых характеристик метрики Холстеда были получены следующие значения:

- 1) $N = 171$;
- 2) $n = 28$;
- 3) $HV = 822,06$.

Полученный показатель объема программы по Холстеду находится в пределах нормы.

4.2.3 Цикломатическая метрика Мак-Кейба

В качестве метрики сложности потока управления (цикломатической сложности) программы была рассмотрена метрика Маккейба.

Программа представляется в виде управляющего ориентированного графа $G(v,e)$, где v – вершины, соответствующие операторам, а e – дуги, соответствующие переходам.

Основной метрикой сложности считается цикломатическая сложность графа программы, или цикломатическое число Маккейба, характеризующее трудоемкость тестирования программы.

Для вычисления цикломатического числа Маккейба $Z(G)$ применяется формула:

$$Z(G) = e - v + 2p,$$

где e – число дуг ориентированного графа G ;

v – число вершин;

p – число компонентов связности графа.

Цикломатическое число Маккейба определяет количество тестовых прогонов программы, необходимых для тестирования по критерию «работает каждая ветвь».

Подсчет цикломатической сложности был проведён на основе управляющего графа прецедента «Просмотр модели клеток» диаграммы вариантов использования (рисунок 40). Показатели для данного графа:

- 1) $e = 8$;

$$2) v = 6;$$

$$3) p = 1.$$

В результате вычисления цикломатического числа метрики Маккейба на основе данных показателей был получено значение 4. Так как цикломатическая сложность метода не должна превышать 10, полученное значение является нормой.

Такой показатель, как индекс сопровождаемости рассчитывается как метрика относительной сложности поддержки, но без слагаемого, учитывающего количество комментариев, и может принимать значения от 0 до 100.

Индекс сопровождаемости кода определяется по формуле:

$$MI = \max(0, (171 - 5.2 * \ln(HV) - 0.23 * CC - 16.2 * \ln(LoC)) * 100/171),$$

где HV (Halstead Volume) – вычислительная сложность;

CC (Cyclomatic Complexity) – структурная сложность кода, т.е. количество различных ветвей в коде;

LoC (Lines of Code) – количество строк кода.

Получены результаты:

$$CC = 4,$$

$$LOC = 82,$$

$$HV = 822,06.$$

В результате вычисления получено значение индекса сопровождаемости кода, равное 38. Таким образом, алгоритм кода просмотр модели клеток имеет приемлемый индекс сопровождаемости.

5 Планирование и расчет бюджета разрабатываемой системы

Управление проектами – это деятельность, в ходе которой намечаются и достигаются цели проекта с учетом рисков и ограничений на трудозатраты, ресурсы и качество. Чтобы достигнуть положительного результата в проектном управлении, нужно иметь заранее составленный план, рационально применять свои знания и инструментальные средства.

При планировании проектов есть несколько основных групп планирования [11]:

- планирование проектных задач;
- планирование расписания проекта;
- планирование команды проекта;
- управление рисками.

При проведении планирования проекта было выделено 29 основных задач. Перечень задач и их виды, предшественники и длительность каждой задачи представлены в таблице 21.

Таблица 21 – Перечень задач проекта

№	Название	Вид задачи	Предшественник	Длительность
1	Начало реализации проекта	Веха	-	-
2	Анализ требований и постановка задачи	Фаза	-	-
3	Анализ программных аналогов	Задача	1	2
4	Постановка задачи	Задача	3	1
5	Анализ функциональных требований	Задача	3	2
6	Выбор средств разработки	Задача	4;5	2
7	Анализ требований завершен	Веха	6	-
8	Проектирование приложений	Фаза	2	-
9	Функциональное моделирование	Задача	7	3
10	Проектирование структур данных	Задача	9	2
11	Проектирование программных средств	Задача	10	4
12	Проектирование пользовательского интерфейса	Задача	10	2
13	Проектирование завершено	Веха	11;12	-

Продолжение таблицы 21

№	Название	Вид задачи	Предшественник	Длительность
14	Реализация	Фаза	8	-
15	Реализация структур данных	Задача	13	3
16	Реализация пользовательского интерфейса	Задача	15	2
17	Реализация программных средств	Задача	16	11
18	Реализация завершена	Веха	17	-
19	Отладка и тестирование	Фаза	14	-
20	Отладка серверной части	Задача	18	3
21	Структурное тестирование	Задача	20	2
22	Рефакторинг	Задача	21	1
23	Анализ качества кода	Задача	22	2
24	Отладка и тестирование завершена	Веха	23	-
25	Завершение проекта	Фаза	19	-
26	Составление документации	Задача	24	5
27	Сдача в эксплуатацию	Задача	26	1
28	Проект завершен	Веха	27	-
29	Конец проекта	Веха	28	-

Для реализации каждой задачи данного проекта необходимо определить ресурсы, а также их тип и стоимостную оценку. Перечень всех ресурсов проекта представлен в таблице 22.

Таблица 22 – Ресурсы и затраты проекта

№	Название ресурса	Тип ресурса	Затраты		Начисление
			Стандартная ставка	Ставка сверхур.	
1	Программист	Трудовой	20 000 р./мес.	250 р./ч.	Пропорциональное
2	Аналитик	Трудовой	14 000 р./мес.	150 р./ч.	Пропорциональное
3	Тестировщик	Трудовой	16 000 р./мес.	200 р./ч.	Пропорциональное
4	Бумага	Материальный	200 р.	-	Пропорциональное
5	Флэш-карта	Материальный	400 р.	-	Пропорциональное
6	Телефон с AR	Материальный	14 000р.	-	Пропорциональное
7	Компьютер	Материальный	28 000р.	-	Пропорциональное
8	Интернет	Затраты	-	-	Пропорциональное

Далее выделенные ресурсы необходимо распределить по задачам. Распределение ресурсов по задачам приведено в таблице 23.

Таблица 23 – Назначение ресурсов

№	Название	Ресурсы	Единицы
1	Начало реализации проекта	Телефон с AR Компьютер	1 шт. 1 шт.
2	Анализ программных аналогов	Интернет Аналитик	350р 100%
3	Постановка задачи	Программист	50%
4	Анализ функциональных требований	Программист Аналитик Интернет	50% 100% 350р
5	Выбор средств разработки	Интернет Программист	350р 100%
6	Функциональное моделирование	Программист	100%
7	Проектирование структур данных	Программист	100%
8	Проектирование программных средств	Программист	70%
9	Проектирование пользовательского интерфейса	Программист	30%
10	Реализация структур данных	Программист	100%
11	Реализация пользовательского интерфейса	Программист	100%
12	Реализация программных средств	Программист	100%
13	Отладка серверной части	Программист Тестировщик	50% 100%
14	Структурное тестирование	Тестировщик	100%
15	Рефакторинг	Программист	100%
16	Анализ качества кода	Тестировщик Программист	100% 50%
17	Составление документации	Бумага Аналитик Программист Тестировщик Флэш-карта Интернет	1 уп. 100% 100% 100% 1 шт. 350р
18	Сдача в эксплуатацию	Аналитик Программист Тестировщик	100% 100% 100%

Для планирования процесса разработки мобильного приложения с дополненной реальностью для изучения структуры растительных клеток была выбрана среда Microsoft Office Project 2007. Созданный проект имеет дату начала 13.04.20. В него были занесены задачи проекта, проанализированные ресурсы и соответствующие им затраты. Результат внесения ресурсов в проект представлен на рисунке 41.

	1	Название ресурса	Тип	Краткое название	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на исполн.	Начисление	Базовый календарь
1		Программист	Трудовой	П	100%	20 000,00 руб./мес	250,00 руб./ч	0,00 руб.	Пропорциональное	Стандартный
2		Аналитик	Трудовой	А	100%	14 000,00 руб./мес	150,00 руб./ч	0,00 руб.	Пропорциональное	Стандартный
3		Тестировщик	Трудовой	Т	100%	16 000,00 руб./мес	200,00 руб./ч	0,00 руб.	Пропорциональное	Стандартный
4		Бумага	Материальный	Б		300,00 руб.		0,00 руб.	Пропорциональное	
5		Флеш_карта	Материальный	Ф		400,00 руб.		0,00 руб.	Пропорциональное	
6		Телефон с AR	Материальный	Тл		14 000,00 руб.		0,00 руб.	Пропорциональное	
7		Компьютер	Материальный	К		28 000,00 руб.		0,00 руб.	Пропорциональное	
8		Интернет	Затраты	И					Пропорциональное	

Рисунок 41 – Ресурсы проекта

Чтобы задача могла быть выполнена, для нее выделяются ресурсы материальные (оборудование) и трудовые (сотрудники). Выделение ресурса задаче называется назначением [12]. Результаты назначений ресурсов в среде Microsoft Office Project представлены на рисунке 42.

	1	Название задачи	Длительность	Начало	Окончание	Предш.	Названия ресурсов
1	✓	Начало реализации проекта	0 дней	Пн 13.04.20	Пн 13.04.20		Телефон с AR[0];Компьютер[0]
2	✓	Анализ требований и постановка	6 дней	Пн 13.04.20	Пн 20.04.20		
3	✓	Анализ программных аналогов	2 дней	Пн 13.04.20	Вт 14.04.20	1	Аналитик;Интернет[350,00 руб.]
4	✓	Постановка задачи	1 день	Ср 15.04.20	Ср 15.04.20	3	Программист[50%]
5	✓	Анализ функциональных требований	2 дней	Ср 15.04.20	Чт 16.04.20	3	Программист[50%];Аналитик;Интернет
6	✓	Выбор средств разработки	2 дней	Пт 17.04.20	Пн 20.04.20	4;5	Программист;Интернет[350,00 руб.]
7	✓	Анализ требований завершен	0 дней	Пн 20.04.20	Пн 20.04.20	6	
8	✓	Проектирование приложений	9 дней	Вт 21.04.20	Пт 01.05.20	2	
9	✓	Функциональное моделирование	3 дней	Вт 21.04.20	Чт 23.04.20	7	Программист
10	✓	Проектирование структур данных	2 дней	Пт 24.04.20	Пн 27.04.20	9	Программист
11	✓	Проектирование программных средств	4 дней	Вт 28.04.20	Пт 01.05.20	10	Программист[70%]
12	✓	Проектирование пользовательского интерфейса	2 дней	Вт 28.04.20	Ср 29.04.20	10	Программист[30%]
13	✓	Проектирование завершено	0 дней	Пт 01.05.20	Пт 01.05.20	11;12	
14	✓	Реализация	16 дней	Пн 04.05.20	Пн 25.05.20	8	
15	✓	Реализация структур данных	3 дней	Пн 04.05.20	Ср 06.05.20	13	Программист
16	✓	Реализация пользовательского интерфейса	2 дней	Чт 07.05.20	Пт 08.05.20	15	Программист
17	✓	Реализация программных средств	11 дней	Сб 09.05.20	Пн 25.05.20	16	Программист
18	✓	Реализация завершена	0 дней	Пн 25.05.20	Пн 25.05.20	17	
19	✓	Отладка и тестирование	8 дней	Вт 26.05.20	Чт 04.06.20	14	
20	✓	Отладка серверной части	3 дней	Вт 26.05.20	Чт 28.05.20	18	Программист[50%];Тестировщик
21	✓	Структурное тестирование	2 дней	Пт 29.05.20	Пн 01.06.20	20	Тестировщик
22	✓	Рефакторинг	1 день	Вт 02.06.20	Вт 02.06.20	21	Программист
23	✓	Анализ качества кода	2 дней	Ср 03.06.20	Чт 04.06.20	22	Программист[50%];Тестировщик
24	✓	Отладка и тестирование завершено	0 дней	Чт 04.06.20	Чт 04.06.20	23	
25	✓	Завершение проекта	6 дней	Пт 05.06.20	Пт 12.06.20	19	
26	✓	Составление документации	5 дней	Пт 05.06.20	Чт 11.06.20	24	Аналитик;Бумага[1];Интернет[350,00 руб.]
27	✓	Сдача в эксплуатацию	1 день	Пт 12.06.20	Пт 12.06.20	26	Аналитик;Программист;Тестировщик
28	✓	Проект завершен	0 дней	Пт 12.06.20	Пт 12.06.20	27	
29	✓	Конец проекта	0 дней	Пт 12.06.20	Пт 12.06.20	28	

Рисунок 42 – Назначение ресурсов задачам

После внесения в проект поставленных задач с распределением по датам реализации и ресурсам среда Microsoft Office Project автоматически построит диаграмму Ганта, которая иллюстрирует календарный график

выполнения работ по проекту. Диаграмма Ганта представлена на рисунке 43.

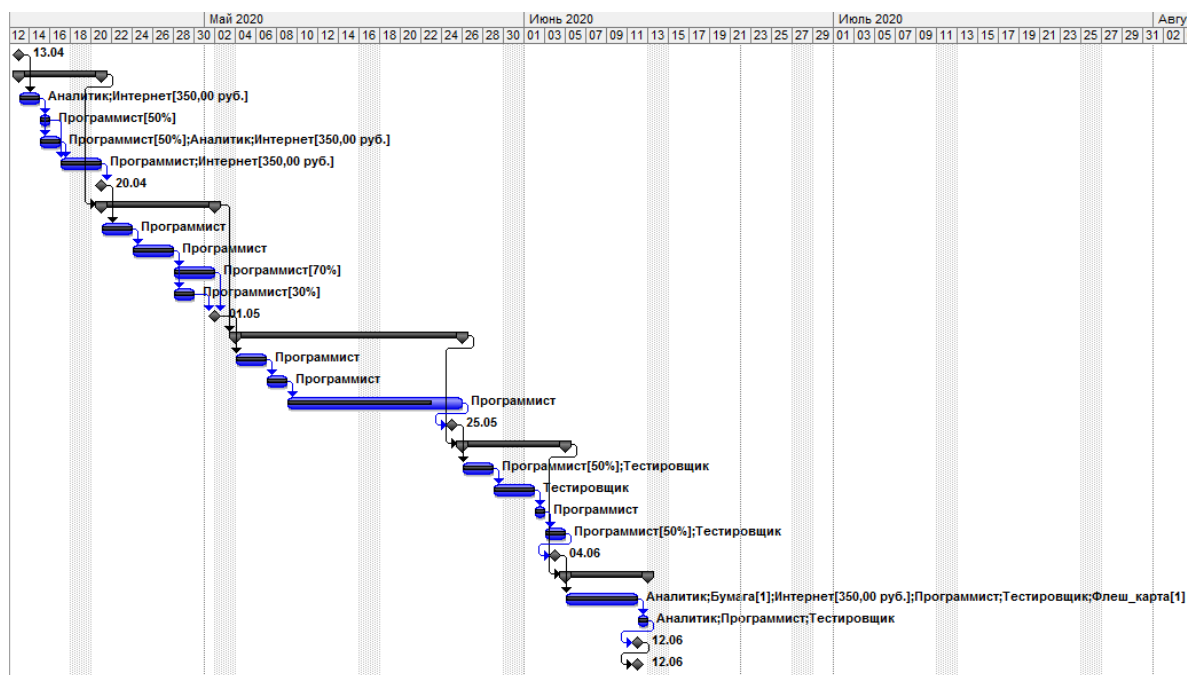


Рисунок 43 – Диаграмма Ганта

При помощи Microsoft Office Project был сформирован обзорный отчет проекта планирования разработки мобильного приложения с дополненной реальностью для изучения структуры растительных клеток. В этом отчете содержатся данные о задаче и количестве затраченных средств на ее выполнение. Обзорный отчет проекта представлен на рисунке 44.

Даты			
Начало:	Пн 13.04.20	Окончание:	Пт 12.06.20
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	Пн 13.04.20	Факт. окончание:	НД
Отклон. начала:	0 дней	Отклон. окончания:	0 дней
Длительность			
Текущий план:	45 дней	Оставшаяся:	0,94 дней
Базовый план:	0 дней?	Фактическая:	44,06 дней
Отклонение:	45 дней	Процент завершения	98%
Трудозатраты			
Текущий план:	483,2 ч	Оставшиеся:	8 ч
Базовый план:	0 ч	Фактические:	475,2 ч
Отклонение:	483,2 ч	Процент завершения	98%
Затраты			
Текущий план:	56 900,00 руб.	Оставшиеся:	2 400,00 руб.
Базовый план:	0,00 руб.	Фактические:	54 500,00 руб.
Отклонение:	56 900,00 руб.		
Состояние задачи		Состояние ресурса	
Еще не начатые задачи:	0	Трудовые ресурсы:	4
Выполняющиеся задачи:	2	Трудовые ресурсы с прев. дост:	0
Завершенные задачи:	27	Материальные ресурсы:	4
Итого задач:	29	Итого ресурсов:	8

Рисунок 44 – Обзорный отчет проекта

В этом отчете содержатся данные о дате начала и конца проекта, длительность и процент завершения всех задач верхнего уровня. Так же построенной таблице отчёта содержатся сведения о затратах для всех задач верхнего уровня. Срок реализации проекта составил 45 рабочих дней с учетом выходных дней и праздников, а бюджет проекта – 56 900 рублей. Каждая задача успешно выполнена в срок и завершена на 100 %.

Также Microsoft Office Project позволил построить отчет по движению денежных средств. Отчет по затратам представлен на рисунке 45.

	2020	2021	Итого
Начало реализации проекта			
Анализ требований и постановка задачи			
Анализ программных аналогов	1 750,00 руб.		1 750,00 руб.
Постановка задачи	500,00 руб.		500,00 руб.
Анализ функциональных требований	2 750,00 руб.		2 750,00 руб.
Выбор средств разработки	2 350,00 руб.		2 350,00 руб.
Анализ требований завершен			
Проектирование приложений			
Функциональное моделирование	3 000,00 руб.		3 000,00 руб.
Проектирование структур данных	2 000,00 руб.		2 000,00 руб.
Проектирование программных средств	2 800,00 руб.		2 800,00 руб.
Проектирование пользовательского интерфейса	600,00 руб.		600,00 руб.
Проектирование завершено			
Реализация			
Реализация структур данных	3 000,00 руб.		3 000,00 руб.
Реализация пользовательского интерфейса	2 000,00 руб.		2 000,00 руб.
Реализация программных средств	11 000,00 руб.		11 000,00 руб.
Реализация завершена			
Отладка и тестирование			
Отладка серверной част	3 900,00 руб.		3 900,00 руб.
Структурное тестирование	1 600,00 руб.		1 600,00 руб.
Рефакторинг	1 000,00 руб.		1 000,00 руб.
Анализ качества кода	2 600,00 руб.		2 600,00 руб.
Отладка и тестирование завершена			
Завершение проекта			
Составление документации	13 550,00 руб.		13 550,00 руб.
Сдача в эксплуатацию	2 500,00 руб.		2 500,00 руб.
Проект завершен			
Конец проекта			
Итого	56 900,00 руб.		56 900,00 руб.

Рисунок 45 – Отчет о движении денежных средств

Кроме того был представлен график движения денежных средств по неделям. График по затратам представлен на рисунке 46.

Отчет о движении денежных средств

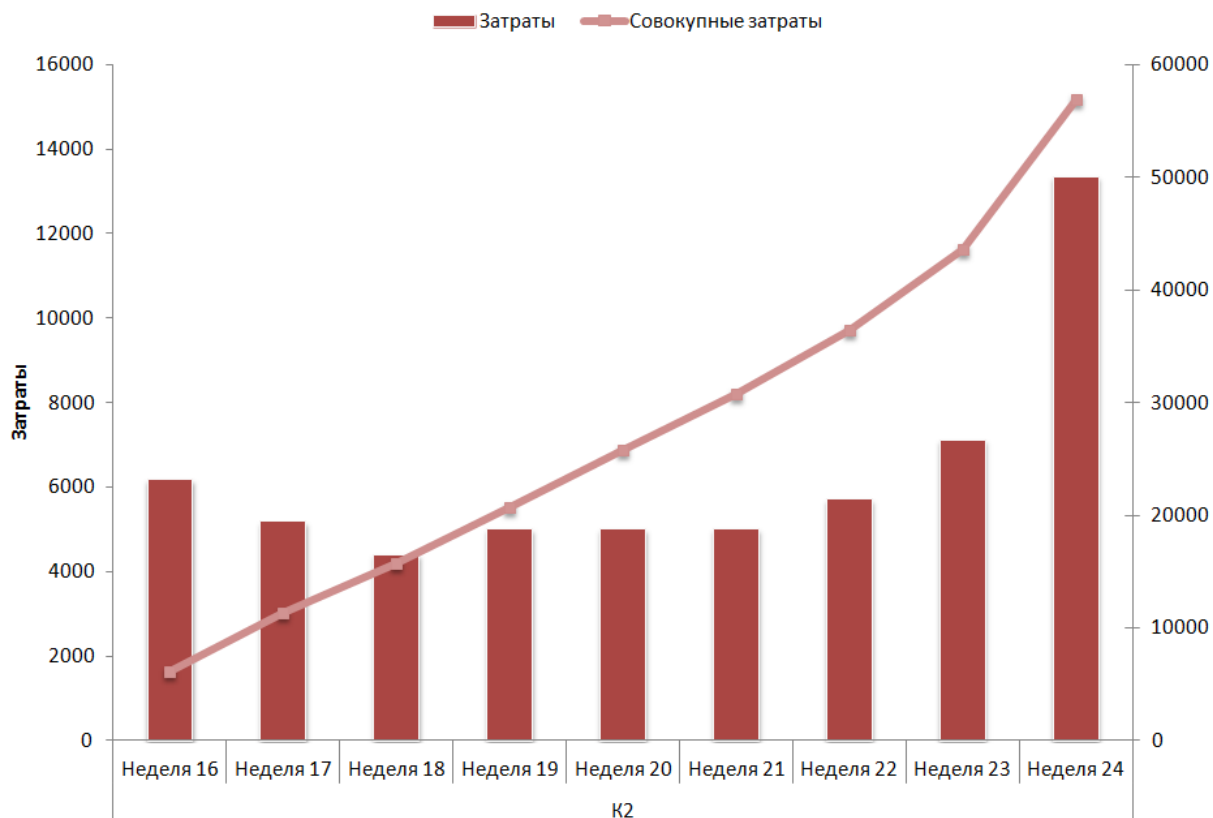


Рисунок 46 – График о движении денежных средств по неделям

На данном графике наглядно показано движение денежных средств по неделям. Линия показывает возрастание общих затрат в течение каждой недели. Как видно последние недели реализации проекта наиболее затратные, так как все работники были задействованы в течение последней недели.

Исходя из приведенных отчетов, а так же при помощи оценки бюджета и сроков проекта можно сделать вывод, что разработка мобильного приложения с дополненной реальностью для изучения структуры растительных клеток является экономически выгодным.

Заключение

В рамках выпускной квалификационной работы разработаны программные средства мобильного приложения с дополненной реальностью для изучения структуры растительных клеток.

Проведено функциональное моделирование в нотации IDEF0 и, при помощи нотации IDEF1X проведено проектирование структур данных.

Разработанное приложение имеет клиент-серверную архитектуру. Для клиентской части используется среда программирования Android Studio, а серверная часть с помощью декларированного языка PostgreSQL и ПО Net Beans IDE.

На этапе анализа качества программного обеспечения проведено структурное тестирование и определены метрики для разрабатываемого программного кода. На основе анализа результатов тестирования можно сделать вывод, что мобильное приложение с дополненной реальностью для изучения структуры растительных клеток функционирует корректно.

В ходе выполнения работы произведено планирование и оценка бюджета программных средств в среде Microsoft Office Project. На основании определенных задач и ресурсов проекта построена диаграмма Ганта. Срок выполнения проекта составил 45 дней, а бюджет – 56 900 рублей. Анализ сроков и бюджета свидетельствует о том, что разработка данного проекта является экономически выгодной.

В ходе реализации выполнены все намеченные в начале задачи. Разработка и написание работы были закончены в установленные сроки.

Таким образом, был разработан программный продукт, отвечающий всем требованиям технического задания. В отличие от аналогов, данное приложение позволяет автоматизировать контроль знаний за счет наличия подсистемы тестирования, что способствует повышению дидактической эффективности процесса обучения. Разработанные программные средства внедрены в международную образовательную платформу Experiment.

Список использованных источников

1. Гриншкун, А. В. Технология дополненной реальности как объект изучения и средство обучения в курсе информатики основной школы: дис. канд. пед. наук: 13.00.02 – М., 2018. – 219 с.
2. Полат, Е.С. Новые педагогические и информационные технологии в системе образования: учебное пособие / Под ред. Е.С. Полат. – М.: Академия, 2001. – 272 с.
3. Като, Х. Манипуляция виртуальными объектами в настольной среде AR / Х. Като, М. Биллингхерст, И. Поупырев, К. Имамото, К. Тачибана // ISAR'00. – 2000. – С.111-119.
4. Гультяев, А.К. Проектирование и дизайн пользовательского интерфейса/ А.К. Гультяев, В.А. Машин: учебное пособие. – СПб.: Корона-Принт, 2007. – 239 с.
5. Прахов, А.А. Blender: 3D-моделирование и анимация. Руководство для начинающих (+ дистрибутив) / А.А. Прахов. – СПб.: БХВ–Петербург, 2009. – 272 с.
6. Маклаков, С. В. Моделирование бизнес-процессов с BPwin 4.0. / С.В. Маклаков. – Москва: ДИАЛОГ-МИФИ, 2002. – 218 с.
7. Проектирование информационных систем с СА ERwin Modeling Suite 7.3: учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154 с.
8. Леоненков, А.В. Самоучитель UML2 / А.В. Леоненков [Электронный ресурс]. – URL: <https://www.e-reading.club/book.php?book=33640> (дата обращения: 23.05.2020).
9. Степанченко, И.В. Методы тестирования программного обеспечения: Учебное пособие / под ред. Л. В. Попова, М. А. Пчелинцена – Волгоград: Изд-во ВолгГТУ, 2006. – 74 с.
10. Майерс, Г. Искусство тестирования программ / Пер. с англ. под ред. Б.А. Позина. – Москва: Финансы и статистика, 1982. – 176 с.
11. Шкрыль, А.А. MS Project 2007: современное управление

проектами. / А.А. Шкрыль. – СПб.: БХВ-Петербург, 2007. – 256 с.

12. Брусенцова Т. П., Смелов В. В. Управление проектами в Microsoft Project: учеб.-метод. пособие для студентов по направлению специальности «Информационные системы и технологии (издательско-полиграфический комплекс)» / – Минск: БГТУ, 2011. – 160 с.

Приложение А
ГЛОССАРИЙ
(справочное)

База данных (БД) – это информативная модель, позволяющая организованно поддерживать данные о группе объектов, имеющих один и тот же набор параметров.

UML (Unified Modeling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

Программное обеспечение (ПО) – совокупность программ, обеспечивающих функционирование компьютеров и решение с их помощью задач предметных областей.

Дополненная реальность (augmented reality, AR) – это среда, в реальном времени дополняющая физический мир, каким мы его видим, цифровыми данными с помощью каких-либо устройств – планшетов, смартфонов или других, и программной части.

Play Маркет (Google Play, Android Market) – магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения.

Blender – профессиональное свободное и открытое программное обеспечение для создания трёхмерной компьютерной графики, включающее в себя средства моделирования, анимации, рендеринга, постобработки и монтажа видео со звуком, компоновки с помощью «узлов», а также для создания интерактивных игр.

Приложение Б

**ЛИСТИНГ СКРИПТА СОЗДАННОЙ
БАЗЫ ДАННЫХ НА ЯЗЫКЕ POSTGRESQL
(рекомендуемое)**

```

-- PostgreSQL database dump

-- Dumped from database version 10.12 (Ubuntu 10.12-0ubuntu0.18.04.1)
-- Dumped by pg_dump version 10.12 (Ubuntu 10.12-0ubuntu0.18.04.1)

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

CREATE EXTENSION IF NOT EXISTS plpgsql WITH SCHEMA pg_catalog;

COMMENT ON EXTENSION plpgsql IS 'PL/pgSQL procedural language';

CREATE EXTENSION IF NOT EXISTS "uuid-oss" WITH SCHEMA public;

COMMENT ON EXTENSION "uuid-oss" IS 'generate universally unique identifiers (UUIDs)';

SET default_tablespace = '';
SET default_with_oids = false;

CREATE TABLE public.cell (
    uuid uuid DEFAULT public.uuid_generate_v4() NOT NULL,
    cell_name character varying(35),
    definition character varying(250),
    updt timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
    updu uuid,
    ver integer,
    del_mark integer DEFAULT 0
);

ALTER TABLE public.cell OWNER TO postgres;

CREATE TABLE public.cell_organell (
    uuid uuid DEFAULT public.uuid_generate_v4() NOT NULL,
    cell uuid,
    organell uuid,
    updt timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
    updu uuid,
    ver integer,
    del_mark integer DEFAULT 0
);

ALTER TABLE public.cell_organell OWNER TO postgres;

CREATE TABLE public.organell (
    uuid uuid DEFAULT public.uuid_generate_v4() NOT NULL,
    organell_name character varying(35),
    definition character varying(250),
    updt timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
    updu uuid,
    ver integer,
    del_mark integer DEFAULT 0
);

```

```
ALTER TABLE public.organell OWNER TO postgres;
```

```
CREATE TABLE public.organell_instance (
    uuid uuid DEFAULT public.uuid_generate_v4() NOT NULL,
    cell_organell uuid,
    path character varying,
    updt timestamp without time zone DEFAULT CURRENT_TIMESTAMP,
    updu uuid,
    ver integer,
    del_mark integer DEFAULT 0
);
```

```
ALTER TABLE public.organell_instance OWNER TO postgres;
```

```
COPY public.cell (uuid, cell_name, definition, updt, updu, ver, del_mark)
FROM stdin;
9dabe2cd-6274-44f9-a464-5d78ddc0c48f    Бактерия        \N        2020-03-18
16:12:10.064991 \N        \N        0
c1faf814-59c2-4bcb-9b05-32d3b460fd93    Вирус    \N        2020-03-18
16:35:16.613863 \N        \N        0
91af708a-650d-4a68-86e2-19906fab6d5f    Животная клетка Структурно-
функциональная единица живого организма, способная к делению и обмену с
окружающей средой    2020-03-18 16:12:10.231074    \N        \N        0
6cb8384e-3321-4a4c-b753-d127caf6b6fc    Растительная клетка    Эукариотические
клетки, еденица живого растительного организма, способная к делению и обмену
с окружающей средой    2020-03-18 16:12:10.192308    \N        \N        0
\.
```

```
COPY public.cell_organell (uuid, cell, organell, updt, updu, ver, del_mark)
FROM stdin;
2c6d3929-e2cc-4425-8ea3-938cca7e411d    6cb8384e-3321-4a4c-b753-d127caf6b6fc
4d8082b1-afe6-417e-a726-d686a55b40fc    2020-03-18 16:28:29.278474
\N        \N        0
dc4e05eb-3cae-4a1a-91af-b00e8d154c39    6cb8384e-3321-4a4c-b753-d127caf6b6fc
4c9e32ce-3659-4203-8f22-ccc82893b4ce    2020-03-18 16:28:29.327252
\N        \N        0
821951ba-f92e-47c7-9b36-af4bc172dbbf    6cb8384e-3321-4a4c-b753-d127caf6b6fc
062a400e-70f3-4349-9205-040f30f63a7a    2020-03-18 16:28:29.37624
\N        \N        0
b2f3ab5c-0061-4f74-b8f5-71f29c386955    6cb8384e-3321-4a4c-b753-d127caf6b6fc
502fbc6d-4cdc-49d4-8bf7-cad01f5def33    2020-03-18 16:28:29.39322
\N        \N        0
7b3e7fd3-fe5d-42c9-9062-9c0dbe120487    6cb8384e-3321-4a4c-b753-d127caf6b6fc
d4a635f4-e7c2-4b4a-8a56-90b7dad73d87    2020-03-18 16:28:29.409922
\N        \N        0
4f54cd76-c05d-425e-9664-b55b827448c1    6cb8384e-3321-4a4c-b753-d127caf6b6fc
ee169a1e-1ea3-4dbf-83db-692505738bf9    2020-03-18 16:28:29.427618
\N        \N        0
096374df-9700-44d1-aaf0-d8349801f0d4    6cb8384e-3321-4a4c-b753-d127caf6b6fc
414d32dd-a383-4501-baa5-9a4b1f13ef1a    2020-03-18 16:28:29.444153
\N        \N        0
73046e7e-5b56-4f19-863c-740aa74b8776    6cb8384e-3321-4a4c-b753-d127caf6b6fc
36b32810-10ae-4f11-832f-6a220cb367d1    2020-03-18 16:28:29.461683
\N        \N        0
bda19f16-411f-45d2-900a-b8d8b6e1853d    6cb8384e-3321-4a4c-b753-d127caf6b6fc
3eb9cab9-938d-46fb-9b09-6ea81d3751aa    2020-03-18 16:28:29.478637
\N        \N        0
c2235eba-4bef-4fa2-91ba-9adbd6eadcf1    6cb8384e-3321-4a4c-b753-d127caf6b6fc
9317848f-703d-49b3-b2d4-3d826035440b    2020-03-18 16:28:29.496178
\N        \N        0
```

0beaa434-ce8b-4fbc-80a4-3624d4313c44 6cb8384e-3321-4a4c-b753-d127caf6b6fc
 690c7e2e-acbc-4b8b-a480-28a927b1312f 2020-03-18 16:28:29.513757
 \N \N 0
 d6a59377-8d2e-47a1-b9f1-34b6935d6799 6cb8384e-3321-4a4c-b753-d127caf6b6fc
 428917b8-2e4f-45fb-a7d4-cb87ebee424b 2020-03-18 16:28:29.530298
 \N \N 0
 618ef9f0-ccd5-4022-b581-9b68cb350e07 6cb8384e-3321-4a4c-b753-d127caf6b6fc
 c3e10a70-b9da-4b92-b283-a4ed899059e7 2020-03-18 16:28:29.547846
 \N \N 0
 be32f47a-fc73-4587-b353-f9bdee718a3c 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
 c8a99cb5-0ecd-4099-9caa-9ed1cfa453c4 2020-03-18 16:32:15.292421
 \N \N 0
 f3db0690-3f5b-46da-81be-0241a756c937 91af708a-650d-4a68-86e2-19906fab6d5f
 690c7e2e-acbc-4b8b-a480-28a927b1312f 2020-03-18 16:32:15.30864
 \N \N 0
 9101a278-8436-4c0e-ad41-2b585c7e37cb 91af708a-650d-4a68-86e2-19906fab6d5f
 428917b8-2e4f-45fb-a7d4-cb87ebee424b 2020-03-18 16:32:15.326178
 \N \N 0
 8454b382-2949-4d58-9ce2-7d3ad46b9e81 91af708a-650d-4a68-86e2-19906fab6d5f
 d4a635f4-e7c2-4b4a-8a56-90b7dad73d87 2020-03-18 16:32:15.342598
 \N \N 0
 fd5199b6-857b-4dc2-9026-4b8098a25c6a 91af708a-650d-4a68-86e2-19906fab6d5f
 ee169a1e-1ea3-4dbf-83db-692505738bf9 2020-03-18 16:32:15.35966
 \N \N 0
 33cc852e-aedb-4379-a62e-0d185269510b 91af708a-650d-4a68-86e2-19906fab6d5f
 414d32dd-a383-4501-baa5-9a4b1f13ef1a 2020-03-18 16:32:15.376584
 \N \N 0
 810af180-9143-4d99-a3c4-2a5be4f626c6 91af708a-650d-4a68-86e2-19906fab6d5f
 3eb9cab9-938d-46fb-9b09-6ea81d3751aa 2020-03-18 16:32:15.394078
 \N \N 0
 6ac6778d-dfe4-45b4-a7df-e79abab693e1 91af708a-650d-4a68-86e2-19906fab6d5f
 36b32810-10ae-4f11-832f-6a220cb367d1 2020-03-18 16:32:15.410804
 \N \N 0
 0ad422e8-a9b9-4f8e-ae08-70b91bbbc096 91af708a-650d-4a68-86e2-19906fab6d5f
 9317848f-703d-49b3-b2d4-3d826035440b 2020-03-18 16:32:15.42853
 \N \N 0
 de21245d-eef9-49fe-83c3-707a4f0b9d7f 91af708a-650d-4a68-86e2-19906fab6d5f
 502fbc6d-4cdc-49d4-8bf7-cad01f5def33 2020-03-18 16:32:15.445653
 \N \N 0
 2ea45c59-05a7-427e-8eb8-f9240f1c1598 91af708a-650d-4a68-86e2-19906fab6d5f
 009dd39b-9f2d-431d-871b-20511a6d3f74 2020-03-18 16:32:15.463321
 \N \N 0
 9b081cbd-5551-4972-b95d-407a5c86802c 91af708a-650d-4a68-86e2-19906fab6d5f
 062a400e-70f3-4349-9205-040f30f63a7a 2020-03-18 16:32:15.480118
 \N \N 0
 13d6326c-a455-434a-a7f1-da0570c01d6c 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
 b93f96f4-52fc-4aa2-b9a8-d1a9787746b9 2020-03-18 16:32:15.497643
 \N \N 0
 b8e2d64a-da90-43f4-892d-928f6ae0d612 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
 062a400e-70f3-4349-9205-040f30f63a7a 2020-03-18 16:32:15.51475
 \N \N 0
 2f245392-00a1-446a-9082-1c253bbe4de4 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
 a57f19c8-52b1-45b1-91b1-a172a5b51f47 2020-03-18 16:32:15.531841
 \N \N 0
 8d10934f-d922-47ed-bbe1-0ba033fbacba 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
 81e3421c-1363-411e-ab2c-f9f1a9305650 2020-03-18 16:32:15.548651
 \N \N 0
 cccfe9ee-3be8-4ae6-9e41-8249402409b6 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
 36b32810-10ae-4f11-832f-6a220cb367d1 2020-03-18 16:32:15.566197
 \N \N 0
 960ac662-a347-46b6-b5e6-0ca91be83795 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
 5f9c3a32-59fd-496c-8f3a-d1059b5bb4ed 2020-03-18 16:32:15.582825
 \N \N 0

```

ab70db47-b372-41d8-a00d-12ac3433ce7a 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
c3e10a70-b9da-4b92-b283-a4ed899059e7 2020-03-18 16:32:15.600496
  \N      \N      0
4b2a29fe-71d2-4e5c-a380-6d2ef407e7f8 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
e8a37d9a-0327-47ad-840f-208993aa1249 2020-03-18 16:32:15.617254
  \N      \N      0
8102eacd-6158-4115-97ff-4297e5d31004 9dabe2cd-6274-44f9-a464-5d78ddc0c48f
f3205271-f302-473d-9803-66805bae5cfa 2020-03-18 16:32:15.634789
  \N      \N      0
ee727793-4eef-41b7-8653-78f931c4eb8d c1faf814-59c2-4bcb-9b05-32d3b460fd93
f13d7895-62b7-4884-bfeb-0cbb4ad90212 2020-03-18 16:36:27.912876
  \N      \N      0
c2691395-3dd2-4048-8850-7d8b3a785d18 c1faf814-59c2-4bcb-9b05-32d3b460fd93
c8a99cb5-0ecd-4099-9caa-9ed1cfa453c4 2020-03-18 16:36:27.934075
  \N      \N      0
1f61e6a6-9cdc-42eb-8d0d-c6092aeb2790 c1faf814-59c2-4bcb-9b05-32d3b460fd93
c2260dd1-44de-4dc3-9a50-09b0142c798c 2020-03-18 16:36:27.950695
  \N      \N      0
9b65d816-07a9-4330-ac3a-2c7256245bf2 c1faf814-59c2-4bcb-9b05-32d3b460fd93
7e369e9f-5df7-485c-b980-d3b5770fa685 2020-03-18 16:36:27.968256
  \N      \N      0
8596d9d5-18cb-4ce1-959f-adf368654546 c1faf814-59c2-4bcb-9b05-32d3b460fd93
41b748cc-374f-4dca-923f-30bc47125d9b 2020-03-18 16:36:27.984892
  \N      \N      0
\.
```

```

COPY public.organell (uuid, organell_name, definition, updt, updu, ver,
del_mark) FROM stdin;
```

```

c3e10a70-b9da-4b92-b283-a4ed899059e7 Клеточная стенка \N 2020-03-
18 16:17:35.80005 \N \N 0
690c7e2e-acbc-4b8b-a480-28a927b1312f Плазматическая мембрана \N
2020-03-18 16:17:35.837694 \N \N 0
36b32810-10ae-4f11-832f-6a220cb367d1 Рибосома \N 2020-03-18
16:17:35.940812 \N \N 0
c8a99cb5-0ecd-4099-9caa-9ed1cfa453c4 ДНК \N 2020-03-18
16:17:36.097581 \N \N 0
b93f96f4-52fc-4aa2-b9a8-d1a9787746b9 Фимбрия \N 2020-03-18
16:17:36.11507 \N \N 0
a57f19c8-52b1-45b1-91b1-a172a5b51f47 Плазмида \N 2020-03-18
16:17:36.14941 \N \N 0
81e3421c-1363-411e-ab2c-f9f1a9305650 Мезосома \N 2020-03-18
16:17:36.165992 \N \N 0
e8a37d9a-0327-47ad-840f-208993aa1249 Капсула \N 2020-03-18
16:17:36.200737 \N \N 0
f3205271-f302-473d-9803-66805bae5cfa Жгутик \N 2020-03-18
16:17:36.218061 \N \N 0
f13d7895-62b7-4884-bfeb-0cbb4ad90212 Гликопротеид \N 2020-03-18
16:17:36.235071 \N \N 0
c2260dd1-44de-4dc3-9a50-09b0142c798c Капсид \N 2020-03-18
16:17:36.252392 \N \N 0
7e369e9f-5df7-485c-b980-d3b5770fa685 Внешняя оболочка \N 2020-03-
18 16:17:36.269216 \N \N 0
41b748cc-374f-4dca-923f-30bc47125d9b Внутренняя оболочка \N 2020-03-
18 16:17:36.286857 \N \N 0
428917b8-2e4f-45fb-a7d4-cb87ebee424b Митохондрия Энергитическая станция
клетки 2020-03-18 15:15:40.289946 \N \N 0
d4a635f4-e7c2-4b4a-8a56-90b7dad73d87 Ядро Наиболее важная часть клетки,
в ней находятся хромосомы, которые несут в себе наследственную информацию
2020-03-18 16:17:35.854114 \N \N 0
502fbc6d-4cdc-49d4-8bf7-cad01f5def33 Ядрышко Образует рибосомные
субъединицы 2020-03-18 16:17:35.871744 \N \N 0
5f9c3a32-59fd-496c-8f3a-d1059b5bb4ed Мембрана Защищает клетку от
внешних воздействий и отвечает за попадание в клетку питательных веществ
2020-03-18 16:17:36.183746 \N \N 0
```

```

9317848f-703d-49b3-b2d4-3d826035440b Лизосома Отвечает за клеточное
пищеварение, уничтожение ненужных клетке органоидов, саморазрушение клетки,
секрецию веществ за пределы клетки 2020-03-18 16:17:36.002362 \N
\N 0
3eb9cab9-938d-46fb-9b09-6ea81d3751aa Аппарат ГольджиПереносит и преобразует
белки, производит липиды 2020-03-18 16:17:35.924162 \N \N
0
062a400e-70f3-4349-9205-040f30f63a7a Цитоплазма Объединяет все
органеллы и обеспечивает их взаимодействие 2020-03-18 16:17:36.132044
\N \N 0
009dd39b-9f2d-431d-871b-20511a6d3f74 Центриоль Образуют клеточный
центр, который нужен при делении клетки 2020-03-18 16:17:35.889499
\N \N 0
414d32dd-a383-4501-baa5-9a4b1f13ef1a Шероховатая эндоплазматическая сеть
Отвечает за синтез белков, так как на ее стенках находятся рибосомы
2020-03-18 16:17:36.081591 \N \N 0
4c9e32ce-3659-4203-8f22-ccc82893b4ce Хлоропласт Содержит зеленый
пигмент, называемый хлорофиллом, который поглощает световую энергию для
процесса фотосинтеза 2020-03-18 16:17:36.303399 \N \N 0
eel69a1e-1ea3-4dbf-83db-692505738bf9 Гладкая эндоплазматическая сеть
Отвечает за синтез жиров (липидов) и углеводов 2020-03-18
16:17:36.063392 \N \N 0
4d8082b1-afe6-417e-a726-d686a55b40fc Вакуоль Отвечает за хранение
питательных веществ, детоксикацию и экспорт отходов в клетке 2020-03-18
16:17:35.905902 \N \N 0
\.
```

```

COPY public.organell_instance (uuid, cell_organell, path, updt, updu, ver,
del_mark) FROM stdin;
\.
```

```

ALTER TABLE ONLY public.cell_organell
ADD CONSTRAINT cell_organell_pk PRIMARY KEY (uuid);
```

```

ALTER TABLE ONLY public.cell
ADD CONSTRAINT cell_pk PRIMARY KEY (uuid);
```

```

ALTER TABLE ONLY public.organell_instance
ADD CONSTRAINT organell_instance_pk PRIMARY KEY (uuid);
```

```

ALTER TABLE ONLY public.organell
ADD CONSTRAINT organell_pk PRIMARY KEY (uuid);
```

```

ALTER TABLE ONLY public.cell_organell
ADD CONSTRAINT cell_organell_fk FOREIGN KEY (cell) REFERENCES
public.cell(uuid);
```

```

ALTER TABLE ONLY public.cell_organell
ADD CONSTRAINT cell_organell_fk_1 FOREIGN KEY (organell) REFERENCES
public.organell(uuid);
```

```

ALTER TABLE ONLY public.organell_instance
ADD CONSTRAINT organell_instance_fk FOREIGN KEY (cell_organell)
REFERENCES public.cell_organell(uuid);
```

```
-- PostgreSQL database dump complete
```

Приложение В

ЛИСТИНГ СЕРВЕРНОЙ ЧАСТИ НА ЯЗЫКЕ JAVA
(рекомендуемое)

Листинг В.1 – Код класса Cell.java

```

@Entity
@Table(name = "cell")
@NamedQueries({
    @NamedQuery(name = "Cell.findAll", query = "SELECT c FROM Cell c"),
    @NamedQuery(name = "Cell.findByCellName", query = "SELECT c FROM Cell c
WHERE c.cellName = :cellName"),
    @NamedQuery(name = "Cell.findByDefinition", query = "SELECT c FROM Cell
c WHERE c.definition = :definition"),
    @NamedQuery(name = "Cell.findByUpdt", query = "SELECT c FROM Cell c
WHERE c.updt = :updt"),
    @NamedQuery(name = "Cell.findByVer", query = "SELECT c FROM Cell c WHERE
c.ver = :ver"),
    @NamedQuery(name = "Cell.findByDelMark", query = "SELECT c FROM Cell c
WHERE c.delMark = :delMark"))
public class Cell extends AbstractEntity {

    @Size(max = 35)
    @Column(name = "cell_name")
    private String cellName;
    @Size(max = 100)
    @Column(name = "definition")
    private String definition;
    @JsonbTransient
    @JsonIgnore
    @OneToMany(mappedBy = "cell")
    private List<CellOrganell> cellOrganellCollection;

    public Cell() {}
    public String getCellName() {
        return cellName; }
    public void setCellName(String cellName) {
        this.cellName = cellName;}
    public String getDefinition() {
        return definition; }

    public void setDefinition(String definition) {
        this.definition = definition; }
    @JsonbTransient
    @JsonIgnore
    public List<CellOrganell> getCellOrganellCollection() {
        return cellOrganellCollection; }

    public void setCellOrganellCollection(List<CellOrganell>
cellOrganellCollection) {
        this.cellOrganellCollection = cellOrganellCollection; }
    @Override
    public String toString() {
        return "school.experiment.biocellback.model.Cell[ id=" + getId() + "
]";
    }
}

```

Листинг В.2 – Код класса CellOrganell.java

```

@Entity
@Table(name = "cell_organell")
@NamedQueries({
    @NamedQuery(name = "CellOrganell.findAll", query = "SELECT c FROM
CellOrganell c"),
    @NamedQuery(name = "CellOrganell.findByUpdt", query = "SELECT c FROM
CellOrganell c WHERE c.updt = :updt"),
    @NamedQuery(name = "CellOrganell.findByVer", query = "SELECT c FROM
CellOrganell c WHERE c.ver = :ver"),

```

```

    @NamedQuery(name = "CellOrganell.findByDelMark", query = "SELECT c FROM
CellOrganell c WHERE c.delMark = :delMark"))))
public class CellOrganell extends AbstractEntity {

    @JoinColumn(name = "cell", referencedColumnName = "uuid")
    @ManyToOne
    private Cell cell;
    @JoinColumn(name = "organell", referencedColumnName = "uuid")
    @ManyToOne
    private Organell organell;

    public CellOrganell() {}
    public Cell getCell() {
        return cell; }
    public void setCell(Cell cell) {
        this.cell = cell; }
    public Organell getOrganell() {
        return organell; }

    public void setOrganell(Organell organell) {
        this.organell = organell; }

    @Override
    public String toString() {
        return "school.experiment.biocellback.model.CellOrganell[ id=" +
getId() + " ]";
    }
}

```

Листинг В.3 – Код класса Organell.java

```

@Entity
@Table(name = "organell")
@NamedQueries({
    @NamedQuery(name = "Organell.findAll", query = "SELECT o FROM Organell o"),
    @NamedQuery(name = "Organell.findByOrganellName", query = "SELECT o FROM
Organell o WHERE o.organellName = :organellName"),
    @NamedQuery(name = "Organell.findByDefinition", query = "SELECT o FROM
Organell o WHERE o.definition = :definition"),
    @NamedQuery(name = "Organell.findByUpdt", query = "SELECT o FROM
Organell o WHERE o.updt = :updt"),
    @NamedQuery(name = "Organell.findByVer", query = "SELECT o FROM Organell
o WHERE o.ver = :ver"),
    @NamedQuery(name = "Organell.findByDelMark", query = "SELECT o FROM
Organell o WHERE o.delMark = :delMark"))))
public class Organell extends AbstractEntity {

    @Size(max = 35)
    @Column(name = "organell_name")
    private String organellName;
    @Size(max = 100)
    @Column(name = "definition")
    private String definition;
    @JsonbTransient
    @JsonIgnore
    @OneToMany(mappedBy = "organell")
    private List<CellOrganell> cellOrganellCollection;

    public Organell() {}
    public String getOrganellName() {
        return organellName; }
    public void setOrganellName(String organellName) {
        this.organellName = organellName; }

    public String getDefinition() {
        return definition; }
}

```

```

public void setDefinition(String definition) {
    this.definition = definition; }

@JsonbTransient
@JsonIgnore
public List<CellOrganelle> getCellOrganelleCollection() {
    return cellOrganelleCollection;
}

public void setCellOrganelleCollection(List<CellOrganelle>
cellOrganelleCollection) {
    this.cellOrganelleCollection = cellOrganelleCollection; }

@Override
public String toString() {
    return "school.experiment.biocellback.model.Organelle[ id=" + getId()
+ " ]";
}}

```

Листинг В.4 – Код класса OrganelleInstance.java

```

@Entity
@Table(name = "organelle_instance")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "OrganelleInstance.findAll", query = "SELECT o FROM
OrganelleInstance o"),
    @NamedQuery(name = "OrganelleInstance.findByPath", query = "SELECT o FROM
OrganelleInstance o WHERE o.path = :path"),
    @NamedQuery(name = "OrganelleInstance.findByUpdt", query = "SELECT o FROM
OrganelleInstance o WHERE o.updt = :updt"),
    @NamedQuery(name = "OrganelleInstance.findByVer", query = "SELECT o FROM
OrganelleInstance o WHERE o.ver = :ver"),
    @NamedQuery(name = "OrganelleInstance.findByDelMark", query = "SELECT o
FROM OrganelleInstance o WHERE o.delMark = :delMark"))
public class OrganelleInstance extends AbstractEntity {

    @JoinColumn(name = "cellorganelle", referencedColumnName = "uuid")
    @ManyToOne
    private CellOrganelle cellOrganelle;
    @Size(max = 2147483647)
    @Column(name = "path")
    private String path;

    public OrganelleInstance() {}
    public String getPath() {
        return path; }
    public void setPath(String path) {
        this.path = path; }

    @Override
    public String toString() {
        return "school.experiment.biocellback.model.OrganelleInstance[ id=" +
getId() + " ]";
    }
    public CellOrganelle getCellOrganelle() {
        return cellOrganelle;
    }

    public void setCellOrganelle(CellOrganelle cellOrganelle) {
        this.cellOrganelle = cellOrganelle;
    }
}

```

Листинг В.5 – Код класса CellFacade.java

```

@Stateless

```

```

public class CellFacade extends AbstractFacade<Cell> {

    @PersistenceContext(unitName =
"school.experiment_biocellback_war_1.0PU")
    private EntityManager entityManager;

    @Override
    protected EntityManager getEntityManager() {
        return entityManager; }
    public CellFacade() {
        super(Cell.class);
    }
}

```

Листинг В.6 – Код класса OrganellFacade.java

```

@Stateless
public class OrganellFacade extends AbstractFacade<Organell> {

    @PersistenceContext(unitName = "school.experiment_biocellback_war_1.0PU")
    private EntityManager entityManager;

    @Override
    protected EntityManager getEntityManager() {
        return entityManager;
    }

    public OrganellFacade() {
        super(Organell.class);
    }
}

```

Листинг В.7 – Код класса CellOrganellFacade.java

```

@Stateless
public class CellOrganellFacade extends AbstractFacade<CellOrganell> {
    @PersistenceContext(unitName = "school.experiment_biocellback_war_1.0PU")
    private EntityManager entityManager;

    protected EntityManager getEntityManager() {
        return entityManager;
    }

    public CellOrganellFacade() {
        super(CellOrganell.class);
    }

    public List<CellOrganell> getForCell(UUID id){
        return getEntityManager()
            .createQuery("SELECT co from CellOrganell co where co.cell.id = :id")
            .setParameter("id", id)
            .getResultList();
    }
}

```

Листинг В.8 – Код класса OrganellInstanceFacade.java

```

@Stateless
public class OrganellInstanceFacade extends AbstractFacade<OrganellInstance>
{
    @PersistenceContext(unitName = "school.experiment_biocellback_war_1.0PU")
    private EntityManager entityManager;

    @Override
    protected EntityManager getEntityManager() {
        return entityManager;
    }

    public OrganellInstanceFacade() {
        super(OrganellInstance.class);
    }
}

```

Листинг В.9 – Код класса CellService.java

```

package school.experiment.biocellback.service;
import io.swagger.annotations.Api;
import java.io.IOException;
import java.net.URISyntaxException;
import java.util.List;
import java.util.UUID;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.ejb.Stateless;
import javax.inject.Inject;
import javax.ws.rs.Consumes;
import javax.ws.rs.GET;
import javax.ws.rs.HeaderParam;
import javax.ws.rs.POST;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;

@Stateless
@Path("cell")
@Api(tags = {"cell rest controller"})
public class CellService extends AbstractService {

    @Inject
    private CellFacade cellFacade;
    private Logger logger = Logger.getLogger(CellService.class.getName());

    @GET
    @Produces({MediaType.APPLICATION_JSON})
    public Response cellorganell() {
        try {
            List<Cell> findAll = cellFacade.findAll();
            return Response.ok(findAll).build();
        } catch (Exception e) {

            logger.log(Level.SEVERE, null, e);
            return Response.serverError().entity(new
StatusMessage(e)).build();
        } }

    @GET
    @Path("{id}")
    @Produces({MediaType.APPLICATION_JSON})
    public Response getById(@PathParam("id") UUID id) {
        try {
            Cell result = cellFacade.find(id);
            return Response.ok(result).build();
        } catch (Exception e) {
            logger.log(Level.SEVERE, null, e);
            return Response.serverError().entity(new
StatusMessage(e)).build();}
    }
}

```

Листинг В.10 – Код класса OrganellService.java

```

@Stateless
@Path("organell")
@Api(tags = {"organell rest controller"})
public class OrganellService extends AbstractService {

    @Inject
    private OrganellFacade OrganellFacade;
    private Logger logger =
Logger.getLogger(OrganellService.class.getName());

```

```

@GET
@Produces({MediaType.APPLICATION_JSON})
public Response cellorganell() {
    try {
        List<Organell> findAll = OrganellFacade.findAll();
        return Response.ok(findAll).build();
    } catch (Exception e) {
        logger.log(Level.SEVERE, null, e);
        return Response.serverError().entity(new
StatusMessage(e)).build();
    } }

@GET
@Path("/{id}")
@Produces({MediaType.APPLICATION_JSON})
public Response getById(@PathParam("id") UUID id) {
    try {
        Organell result = OrganellFacade.find(id);
        return Response.ok(result).build();
    } catch (Exception e) {
        logger.log(Level.SEVERE, null, e);
        return Response.serverError().entity(new
StatusMessage(e)).build();}
}}

```

Листинг В.11 – Код класса CellOrganellService.java

```

@Stateless
@Path("cellorganell")
@Api(tags = {"cell organell rest controller"})
public class CellOrganellService extends AbstractService {

    @Inject
    private CellOrganellFacade cellOrganellFacade;
    private Logger logger =
Logger.getLogger(CellOrganellService.class.getName());

    @GET
    @Produces({MediaType.APPLICATION_JSON})
    public Response cellorganell() {
        try {
            List<CellOrganell> findAll = cellOrganellFacade.findAll();
            return Response.ok(findAll).build();
        } catch (Exception e) {
            logger.log(Level.SEVERE, null, e);
            return Response.serverError().entity(new
StatusMessage(e)).build();
        } }

    @GET
    @Path("cellorganell/{id}")
    @Produces({MediaType.APPLICATION_JSON})
    public Response getById(@PathParam("id") UUID id) {
        try {
            CellOrganell result = cellOrganellFacade.find(id);
            return Response.ok(result).build();
        } catch (Exception e) {
            logger.log(Level.SEVERE, null, e);
            return Response.serverError().entity(new
StatusMessage(e)).build();
        } }

    @GET
    @Path("getforcell/{id}")
    @Produces({MediaType.APPLICATION_JSON})
    public Response getForCell(@PathParam("id") UUID id) {

```

```

        try {
            List<CellOrganell> findorg = cellOrganellFacade.getForCell(id);
            return Response.ok(findorg).build();
        } catch (Exception e) {
            logger.log(Level.SEVERE, null, e);
            return Response.serverError().entity(new
StatusMessage(e)).build();
        }
    }
}

```

Листинг В.12 – Код класса OrganellInstanceService.java

```

@Stateless
@Path("organellinstance")
@Api(tags = {"organell instance rest controller"})
public class OrganellInstanceService extends AbstractService {

    @Inject
    private OrganellInstanceFacade OrganellInstanceFacade;
    private Logger logger =
Logger.getLogger(CellOrganellService.class.getName());

    @GET
    @Produces({MediaType.APPLICATION_JSON})
    public Response cellorganell() {
        try {
            List<OrganellInstance> findAll =
OrganellInstanceFacade.findAll();
            return Response.ok(findAll).build();
        } catch (Exception e) {
            logger.log(Level.SEVERE, null, e);
            return Response.serverError().entity(new
StatusMessage(e)).build();
        }
    }

    @GET
    @Path("{id}")
    @Produces({MediaType.APPLICATION_JSON})
    public Response getById(@PathParam("id") UUID id) {
        try {
            OrganellInstance result = OrganellInstanceFacade.find(id);
            return Response.ok(result).build();
        } catch (Exception e) {
            logger.log(Level.SEVERE, null, e);
            return Response.serverError().entity(new
StatusMessage(e)).build();
        }
    }
}

```

Листинг В.13 – Код класса Authorization.java

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_authorization);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    if (ContextCompat.checkSelfPermission(Authorization.this,
Manifest.permission.GET_ACCOUNTS) != PackageManager.PERMISSION_GRANTED) {
        if (ActivityCompat.shouldShowRequestPermissionRationale(Authorization.this,
Manifest.permission.GET_ACCOUNTS)) {
            ActivityCompat.requestPermissions(Authorization.this,
new String[]{Manifest.permission.GET_ACCOUNTS},
MY_PERMISSIONS_REQUEST_GET_ACCOUNTS);
            ActivityCompat.requestPermissions(Authorization.this,
new String[]{Manifest.permission.READ_CONTACTS},
MY_PERMISSIONS_REQUEST_GET_ACCOUNTS);
        } else {

```

```

ActivityCompat.requestPermissions(Authorization.this,
    new String[]{Manifest.permission.GET_ACCOUNTS},
    MY_PERMISSIONS_REQUEST_GET_ACCOUNTS);
ActivityCompat.requestPermissions(Authorization.this,
    new String[]{Manifest.permission.READ_CONTACTS},
    MY_PERMISSIONS_REQUEST_GET_ACCOUNTS);
}
} else {
NextLevel();
}}
@Override
public void onRequestPermissionsResult(int requestCode, String
permissions[], int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_GET_ACCOUNTS: {
            if (grantResults.length > 0
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) { NextLevel();
            } else {
                AlertDialog.Builder builder = new AlertDialog.Builder(Authorization.this);
                builder.setTitle("Внимание!")
                    .setMessage("К сожалению, без нужных прав я не могу работать :(")
                    .setCancelable(false)
                    .setNegativeButton("Хорошо",
                        new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog, int id) {
                                dialog.cancel();
                                finishAffinity();
                            }
                        });
                AlertDialog alert = builder.create();
                alert.show();
                return;
            }
        }
        private void NextLevel() {
            AccountManager accountManager = AccountManager.get(Authorization.this);
            Account[] accounts = accountManager.getAccountsByType("experiment_account");
            if (accounts.length != 0) {
                Intent intent = new Intent(Authorization.this, MainActivity.class);
                intent.putExtra("userToken", accountManager.getUserData(accounts[0],
                    "token"));
                startActivity(intent);
                finish();
            } else {
                AlertDialog.Builder builder = new AlertDialog.Builder(Authorization.this);
                builder.setTitle("Внимание!")
                    .setMessage(R.string.link)
                    .setCancelable(false)
                    .setNegativeButton("Хорошо",
                        (dialog, id) -> {
                            dialog.cancel();
                            finishAffinity();
                        });
                AlertDialog alert = builder.create();
                alert.show();
            }
        }
    }
}

```

Приложение Г
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
(рекомендуемое)

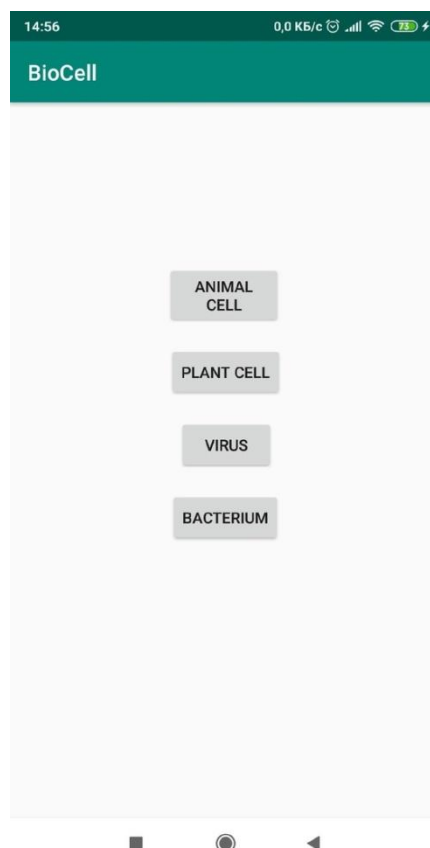


Рисунок Г.1 – Структурное тестирование, тест 1

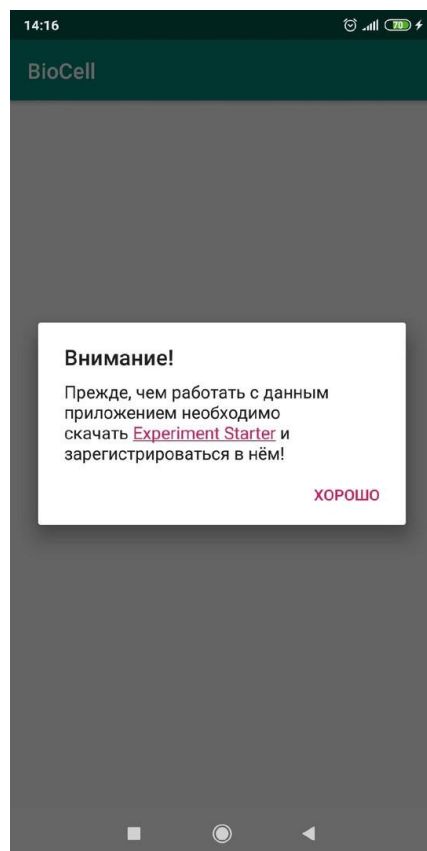


Рисунок Г.2 – Структурное тестирование, тест 2

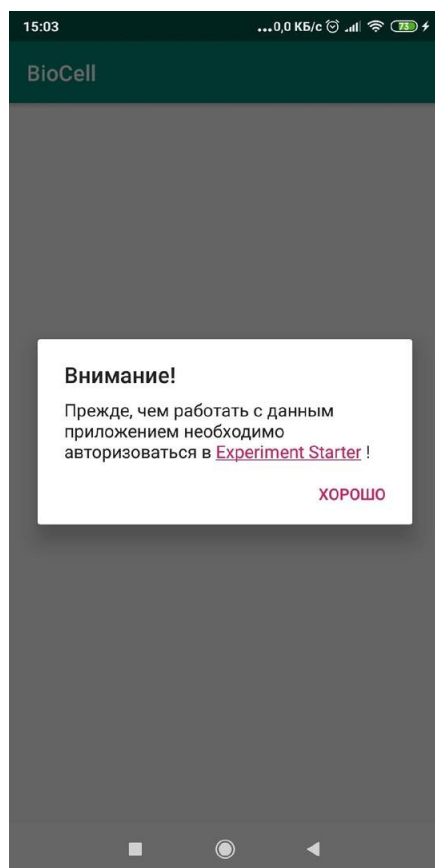


Рисунок Г.3 – Структурное тестирование, тест 3

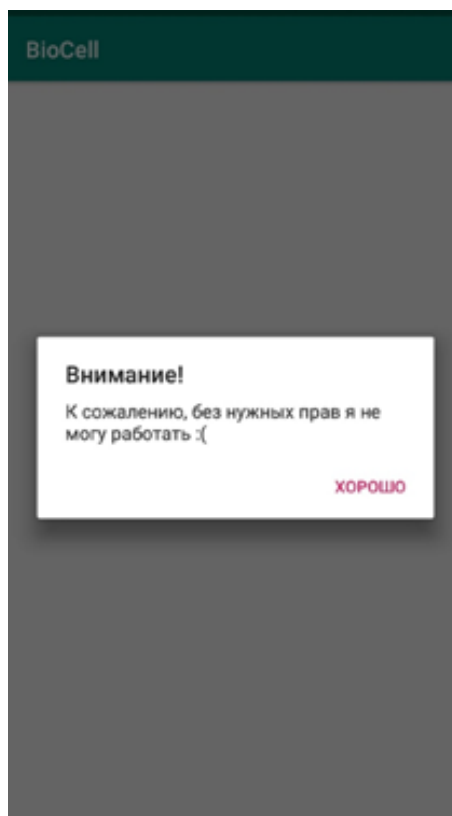


Рисунок Г.4 – Структурное тестирование, тест 4

Приложение Д
АКТ О РЕЗУЛЬТАТАХ ВНЕДРЕНИЯ
(рекомендуемое)

ООО «СЕТАП»site: setupit.org

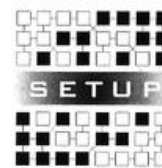
ОГРН: 1177746688416

ИНН: 7704434695

email: info@setupit.org

119435, г. Москва, ул. Пироговская М.,

д.6/4, корп. 2, пом. V КОМ 5

**Акт**

о внедрении результатов
выпускной квалификационной работы
Гусева Дмитрия Владимировича

Мобильное приложение «BioCell» для изучения структуры растительной клетки по курсу биологии с использованием технологии дополненной реальности, разработанное стажером ООО «СЕТАП» Гусевым Д.В. в ходе выпускной квалификационной работы, включено в программный комплекс Experiment и является его неотъемлемой частью. Данное решение разрабатывается в рамках образовательной онлайн-платформы Experiment.

Генеральный директор ООО «СЕТАП»

Хованский Д.Н.

26.06.20