***Р.Р. Газизов, прикладная информатика,***

***3 курс***

***Н.В. Ефанова,   
доцент, кандидат экономических наук***

***ФГБОУ ВО «Кубанский государственный аграрный университет  
имени И.Т. Трубилина»,***

***Российская Федерация***

**Мобильное приложение Ustudent**

В статье рассказывается о создании мобильного приложения для студентов «UStudent». Объясняется актуальность данного проекта, описываются выбранные технологии и подходы. Также обращается внимание на важность ведения документации в процессе разработки, как для облегчения работы самой команды разработчиков, так и для понимания и технической поддержки приложения в дальнейшем.

The article discusses the creation of a mobile application for students called «UStudent». It explains the relevance of this project, describes the chosen technologies and approaches. Attention is also drawn to the importance of documentation during the development process, both for the ease of the development team and for understanding and technical support of the application in the future.

Мы пользуемся мобильными телефонами каждый день. Для кого-то это проблема и даже зависимость, но для кого-то это удобный инструмент для решения повседневных задач по работе, учебе и жизни. Несомненно, нужно стремиться причислять себя ко второй группе лиц, для которой современные технологии — лишь инструмент воплощения идей, а не источник проблем.

Так, для студентов было разработано множество приложений, упрощающих нелегкий процесс обучения. Сегодня многие предпочитают стандартным ежедневникам и тетрадям их цифровые аналоги, а во многих университетах существуют внутренние информационные сервисы, предоставляющие студентам необходимый функционал. Если рассматривать имеющиеся на цифровых площадках приложения, можно отметить одну проблему, относящуюся ко всем продуктам данной предметной области, ­– отсутствие интеграции с университетами.

Устанавливая любое приложение по ведению расписания, мониторингу посещаемости и успеваемости, пользователи должны вносить все дисциплины и настраивать расписание вручную, что уже отталкивает от использования, ведь объем данных достаточно большой. Если учитывать, что в расписании и дисциплинах могут быть корректировки, студенту придется постоянно сверять свои записи с первоисточником во избежание неточностей. Помимо этого, в таких приложениях совсем обделена вниманием роль старосты группы. В связи с этим было принято решение создать собственное мобильное приложение, предоставляющее студентам наиболее востребованный функционал, а также имеющее потенwциал стать универсальным приложением для всех вузов России — UStudent.

На данный момент приложение доступно на ОС Android в цифровом магазине RuStore, а пользоваться им могут студенты Кубанского государственного аграрного университета. В первую очередь было необходимо проверить гипотезу о востребованности функционала студенческого приложения, создав узкоспециализированное, но не требующее сложного для разработки «универсального» модуля.

В рамках дисциплины «Программная инженерия» разработка минимально рабочей версии продукта заняла три месяца. Работа велась по методологии гибкой разработки Agile с использованием Jira в качестве системы управления проектом. После изучения рынка на наличие аналогов и их слабых мест было необходимо сформировать границы проекта, функционал, который будет реализован в течение семестра.

За это время был реализован функционал просмотра расписания и учета посещаемости студентов для старост.

Пользователи могут просматривать расписание по группам, аудиториям и преподавателям, а расписание своей группы доступно даже в режиме офлайн. Немало внимания уделяется и роли старосты в приложении. Так, староста может составлять списки присутствующих или отсутствующих студентов на занятиях, указывать причины отсутствия, выбирать дату и занятие с учетом расписания в этот день и отправлять список преподавателю на почту или в мессенджер. Ниже представлен интерфейс приложения для реализованного функционала (см. рис. 1)

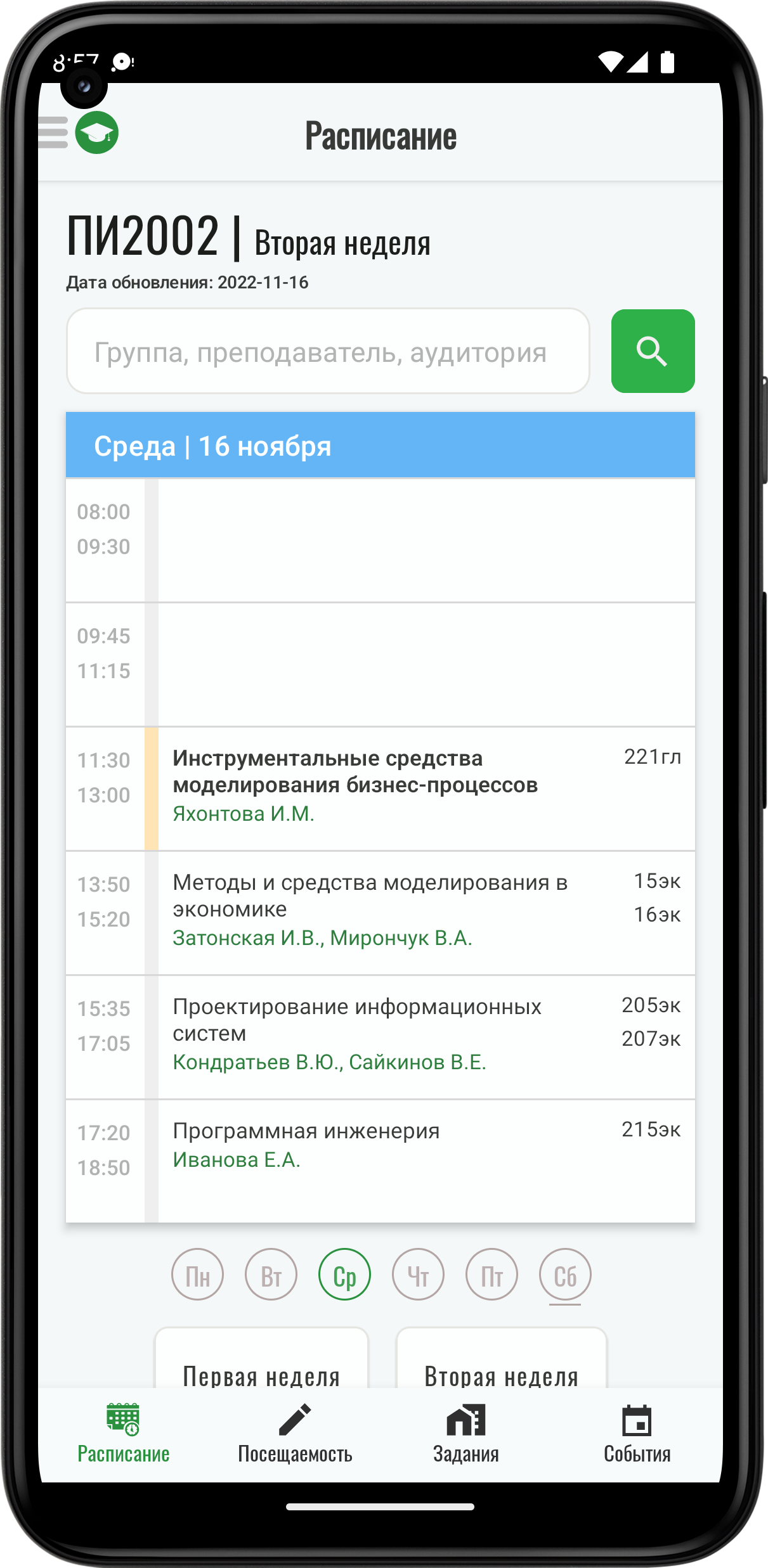
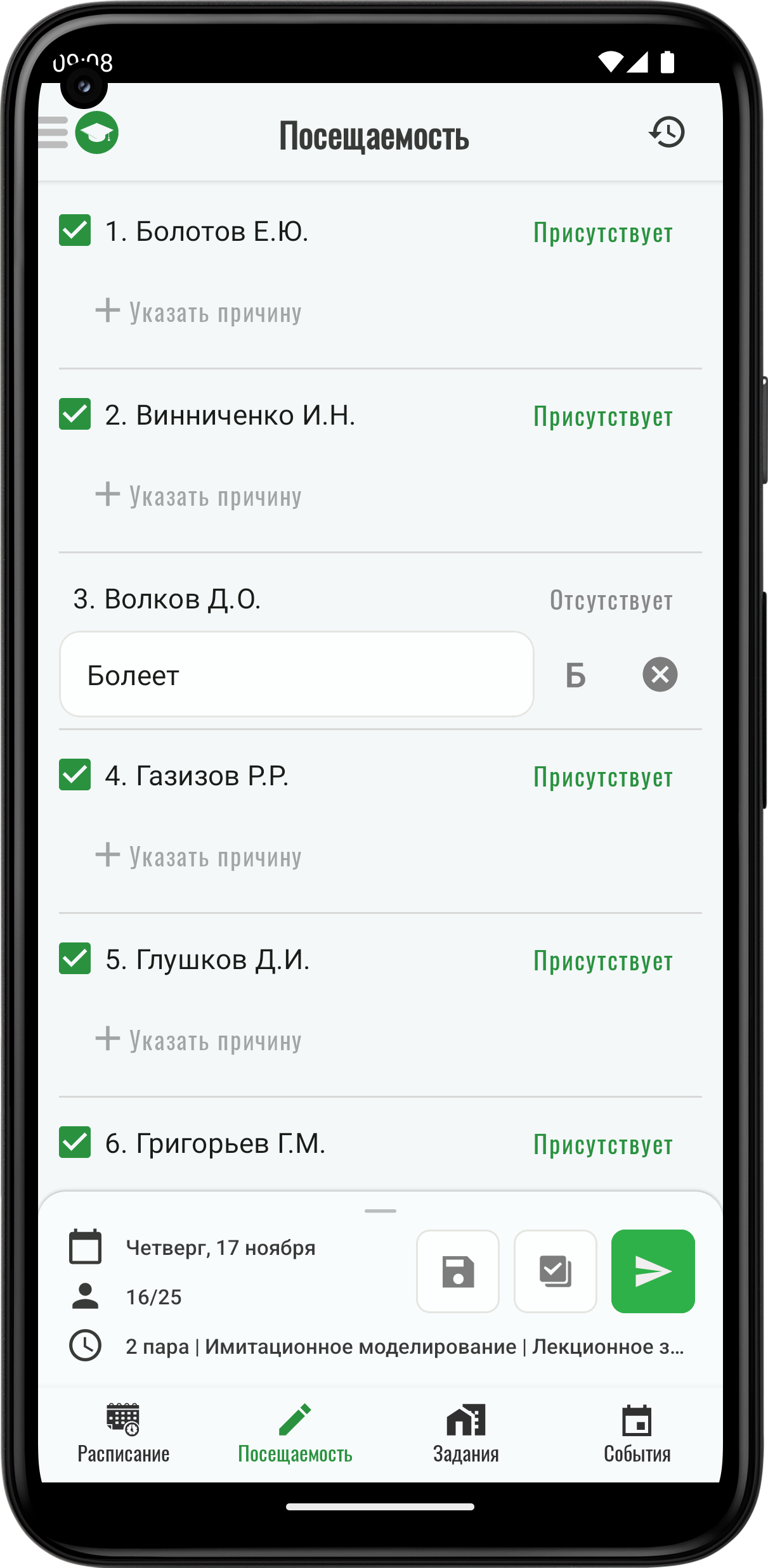
 

Рисунок 1 — интерфейс приложения

Разработка Android приложения осуществлялась на языке программирования Kotlin. В целях обеспечения масштабируемости и корректности работы было принято решение придерживаться принципов «чистой архитектуры». «Чистая архитектура» — это подход к созданию архитектуры ПО, который стремится отделить бизнес-логику от деталей реализации. Это делается с помощью разделения кода на слои, каждый из которых имеет свою область ответственности. В центре чистой архитектуры находится слой бизнес-логики, который не зависит от способа реализации и может быть использован в различных системах. Он связан с другими слоями через интерфейсы, что позволяет легко заменять их реализацию без нарушения бизнес-логики.

Интерфейс занимает центральное место в любом приложении. Для построения экранов использовался декларативный фреймворк по созданию нативного UI — Jetpack Compose. Для связи интерфейса со слоем бизнес-логики использовался архитектурный паттерн MVVM (Model-View-ViewModel). MVVM позволяет разделить логику приложения и сделать ее более тестируемой, повышая гибкость и масштабируемость кода. Он используется в связке с реактивным программированием, что позволяет отображать динамические изменения данных в реальном времени.

Документирование программного проекта — задача не легкая, но обязательная к выполнению. В каждый момент времени нужно понимать, в каком состоянии находится проект. Документация позволяет устранить художественное видение проекта со стороны программиста, наладить работу в команде, а также ясно сформулировать миссию проекта. Помимо этого, документация критически необходима в кроссплатформенных проектах, чтобы пользовательский опыт был одинаково хорош на любых ОС. К независимым от платформы артефактам документации относятся:

* Диаграмма классов БД (см. рисунок 2)

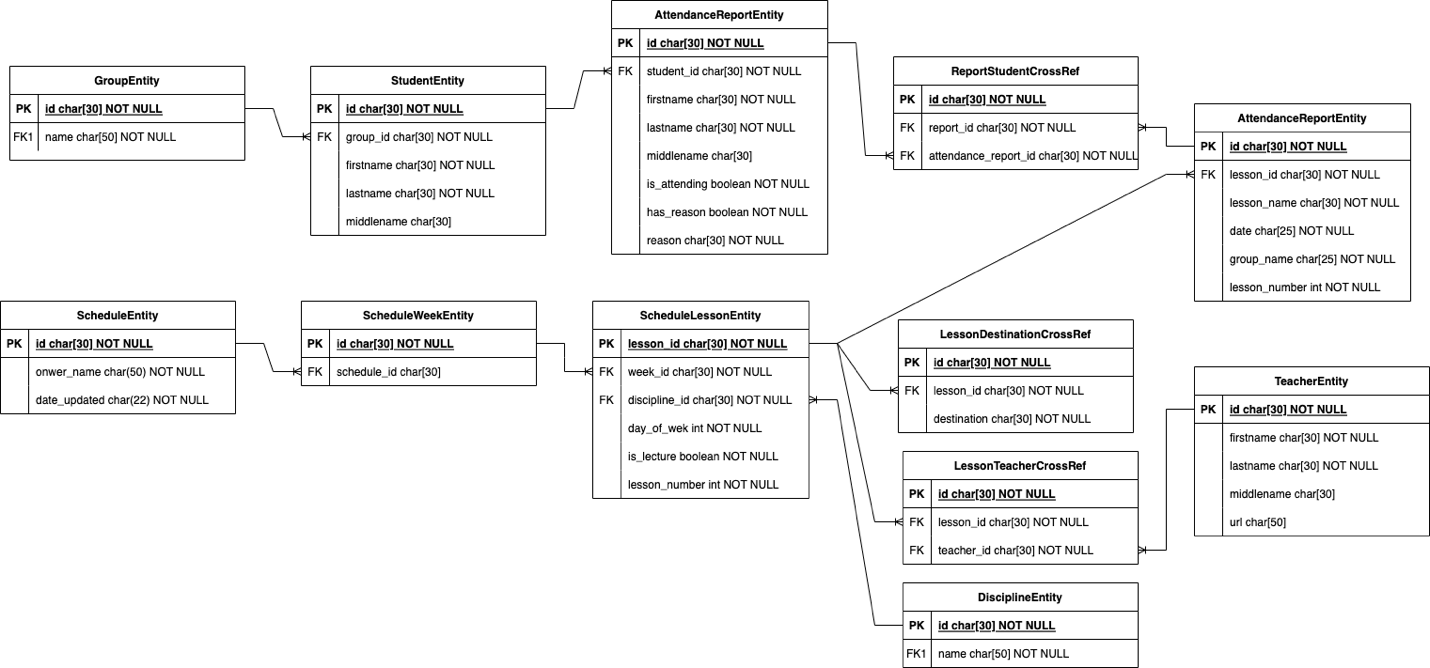


Рисунок 2 — диаграмма классов БД

* схема переходов между экранами (см. рисунок 3)

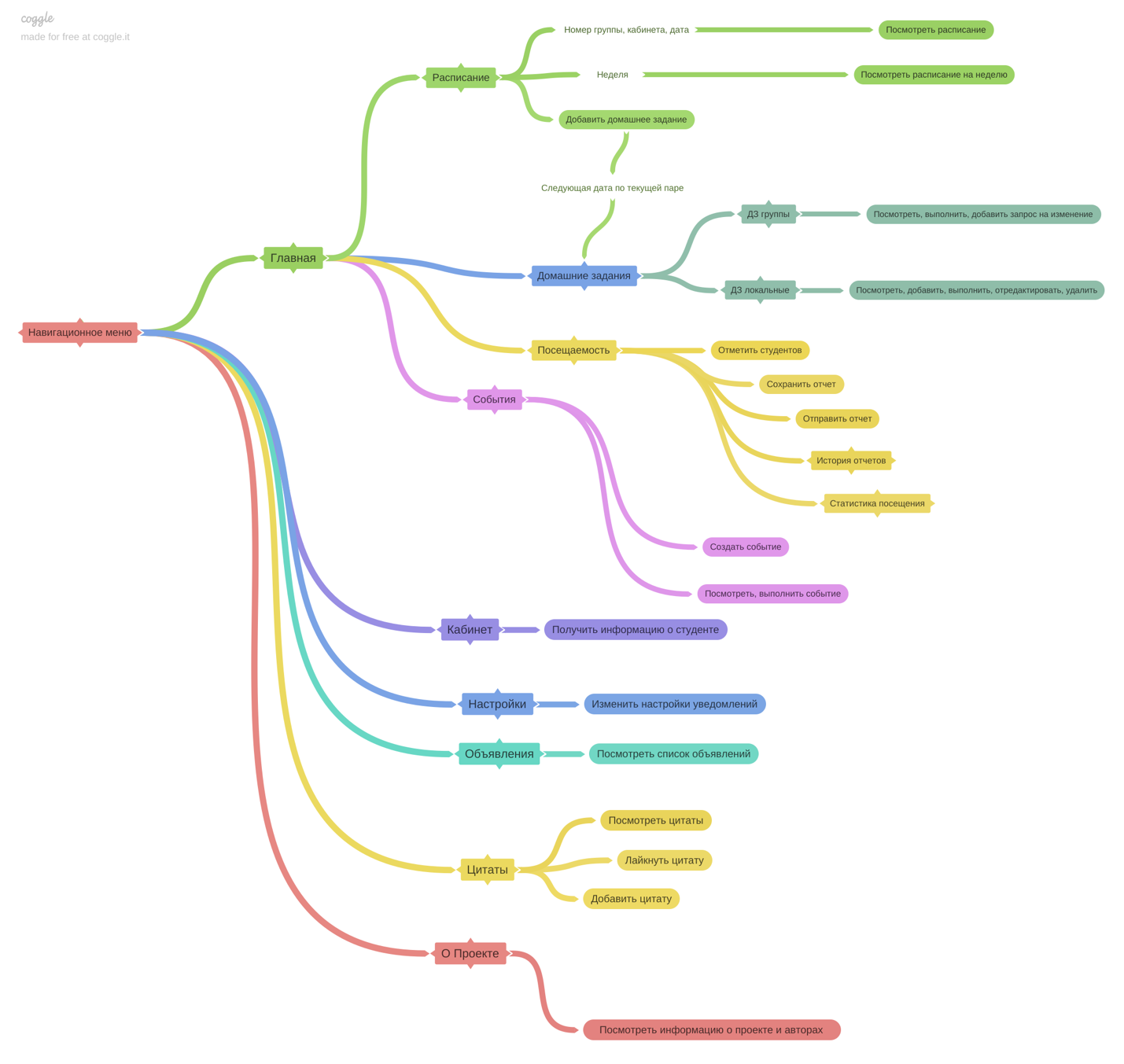


Рисунок 3 — схема переходов

* Прописанные пользовательские сценарии и варианты использования:

|  |  |
| --- | --- |
| Код и Наименование ВИ | 1 Составить списки студентов |
| Краткое описание | Составление отчета по посещаемости на занятии |
| Актеры | Староста |
| Ссылки на другие ВИ | - |
| Предусловия | Группа по умолчанию была установлена.  Пользователь переходит в раздел «Посещаемость» |
| Основной поток | * Пользователь отмечает присутствующих студентов из списка * Пользователь выбирает дату и пару * Система спрашивает, какой список подготовить к отправке * Система отправляет отчет в приложение по выбору пользователя |
| Постусловия | Отчет был отправлен, пользователь возвращается в приложение |
| Альтернативные потоки | - |
| Специальные требования | - |

Составлением документации занимается технический писатель, но программисту также необходимо знать нотации диаграмм, так как они являются ядром проекта, его бизнес-логикой.

Реализация мобильного приложения «UStudent» в рамках дисциплины «Программная инженерия» позволила испытать себя во всех ролях, начиная от архитектора и дизайнера, заканчивая программистом и пользователем. Данный опыт показал, насколько объемным оказался процесс создания программного продукта, с какими трудностями сталкиваются члены команды, и как важна правильная организация рабочего пространства и времени.

Использованные источники

1. Гамма Эрих, Хелм Ричард, Джонсон Роберт, Влиссидес Джон – Паттерны объектно-ориентированного программирования – СПб.: Питер, 2022 – 448 с.: ил. – (Серия «Библиотека программиста»).
2. Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр. : ил.
3. Программная инженерия : учебник / Н. В. Ефанова, Е. А. Иванова. – Краснодар : КубГАУ, 2022. – 174 с.