

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение школы (НОЦ): Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Оптимизация системы навигации мобильного приложения под ОС Android
УДК <u>004.75:004.455-025.26</u>

Студент

Группа	ФИО	Подпись	Дата
8К51	Ворожейкин Вячеслав Александрович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Соколова Вероника Валерьевна	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСТН ШБИП ТПУ	Подопригора Игнат Валерьевич	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП ТПУ	Винокурова Галина Федоровна	К.Т.Н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.04 «Программная инженерия»
 Отделение школы: Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Чердынцев Е.С.
 (Подпись) (Дата)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы (бакалаврской работы, дипломного проекта/работы, магистерской диссертации)
--

Студенту:

Группа	ФИО
8K51	Ворожейкину Вячеславу Александровичу

Тема работы:

Оптимизация системы навигации мобильного приложения под ОС Android	
Утверждена приказом директора (дата, номер)	№ 4454/с от 30.05.2019

Срок сдачи студентом выполненной работы:	19.06.2019
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объектом исследования является существующая система навигации мобильного приложения «Camfrog» под ОС Android, которая подлежит оптимизации.</p>
<p>Перечень подлежащих исследованию, проектированию и разработке вопросов <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Исследование предметной области 2. Проектирование модуля навигации 3. Разработка компонентов нового решения 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 5. Социальная ответственность

<p>Перечень графического материала (с точным указанием обязательных чертежей)</p>	<ol style="list-style-type: none"> 1. Диаграммы в нотации IDEF0 2. Причинно-следственная диаграмма Исикавы 3. Диаграмма вариантов использования в нотации UML 4. Диаграмма в нотации IDEF3 5. Диаграммы потоков данных в нотации DFD 6. Диаграмма моделирования бизнес-процесса в нотации BPMN 7. Диаграмма моделирования бизнес-процесса в нотации EPC 8. Диаграммы классов в нотации UML 9. Пояснительные скриншоты приложения 10. Многоугольник конкурентоспособности 11. Матрица SWOT-анализа 12. Диаграмма Гантта 13. Таблица трудозатрат
--	---

<p>Консультанты по разделам выпускной квалификационной работы (с указанием разделов)</p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Подопригора Игнат Валерьевич</p>
<p>Социальная ответственность</p>	<p>Винокурова Галина Федоровна</p>

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	<p>01.03.2019</p>
--	-------------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
<p>Доцент ОИТ ИШИТР ТПУ</p>	<p>Соколова Вероника Валерьевна</p>	<p>к.т.н.</p>		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
<p>8К51</p>	<p>Ворожейкин Вячеслав Александрович</p>		

Школа: Инженерная школа информационных технологий и робототехники

Направление подготовки: Программная инженерия

Уровень образования: Бакалавр

Отделение школы: Отделение информационных технологий

Период выполнения: осенний / весенний семестр 2018/2019 учебного года

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	19.06.2019
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
03.05.2019	Глава 1. Исследование предметной области	15
10.05.2019	Глава 2. Проектирование модуля навигации	20
17.05.2019	Глава 3. Разработка компонентов нового решения	25
24.05.2019	Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
28.05.2019	Глава 5. Социальная ответственность	20

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Соколова Вероника Валерьевна	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8К51	Ворожейкину Вячеславу Александровичу

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Оклад руководителя команды мобильной разработки – 64527 руб. Оклад научного руководителя – 33664 руб. Оклад тестировщика – 38925 руб. Оклад разработчика – 25287 руб.
2. <i>Нормы и нормативы расходования ресурсов</i>	Премиальный коэффициент 30 %. Коэффициент доплат и надбавок 35 %. Районный коэффициент 30 %. Коэффициент дополнительной заработной платы 13 %. Накладные расходы 16 %.
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на уплату во внебюджетные фонды 28 %

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	-Анализ конкурентных технических решений
2. <i>Планирование и формирование бюджета научных исследований</i>	Формирование плана и графика разработки: - определение структуры работ; - определение трудоемкости работ; - разработка графика Гантта. Формирование бюджета затрат на научное исследование: - материальные затраты; - заработная плата (основная и дополнительная); - отчисления на социальные цели; - накладные расходы.
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	- Определение потенциального эффекта

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценочная карта конкурентных технических решений
2. Матрица SWOT-анализа
3. График Гантта
4. Расчет бюджета затрат

Дата выдачи задания для раздела по линейному графику	01.03.2019
---	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСТН ШБИП ТПУ	Подопригора Игнат Валерьевич	К.Э.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К51	Ворожейкин Вячеслав Александрович		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8К51	Ворожейкину Вячеславу Александровичу

Школа	ИШИТР	Отделение (НОЦ)	Информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Оптимизация системы навигации мобильного приложения под ОС Android	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования – программный модуль навигации мобильного приложения под ОС Android, используемый разработчиками. Рабочее место – часть пространства офисного помещения с персональными рабочим столом и компьютером.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	– Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ – ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. – СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	– Электромагнитные излучения – Недостаток естественного освещения – Эмоциональные перегрузки – Опасность поражения электрическим током
3. Экологическая безопасность:	Загрязнение окружающей среды вследствие использования и утилизации компьютерной техники
4. Безопасность в чрезвычайных ситуациях:	Пожар

Дата выдачи задания для раздела по линейному графику	01.03.2019
--	------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП ТПУ	Винокурова Галина Федоровна	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К51	Ворожейкин Вячеслав Александрович		

Реферат

Выпускная квалификационная работа содержит 97 страниц, 27 рисунков, 18 таблиц, 1 приложение и 26 литературных источников.

Ключевые слова: навигация, оптимизация, ОС Android, окно, переход.

Цель работы: проведение оптимизации системы навигации мобильного приложения для увеличения ее гибкости.

В результате работы была перепроектирована существующая система навигации и разработана новая.

В первой главе представлено описание предметной области, детализация проблемы в существующем решении, обзор возможных решений.

Вторая глава отображает проектирование решения, включающее в себя моделирование вариантов использования, потоков данных, функциональное моделирование и описание бизнес-логики процесса.

В третьей главе представлен результат разработки через описание реализации решения с демонстрацией изменений, а также примеры использования новой системы навигации.

Четвертая глава представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение», в котором отображены потенциал, планирование и эффективность решения.

Пятая глава представляет собой выполненное задание по разделу «Социальная ответственность», в котором рассмотрены организационно-правовые, производственные и экологические аспекты безопасности, а также безопасность в чрезвычайных ситуациях.

Список терминов и сокращений

- 1. Унификация** – приведение к единообразию.
- 2. Видеочат** – онлайн-сервис, позволяющий пользователям Интернета общаться, в дополнение к текстовой переписке, в аудио- и видео-режимах.
- 3. Android** – операционная система для мобильных устройств, созданная компанией Google на основе ОС Linux.
- 4. Activity (операция)** – компонент мобильного приложения, который предоставляет экран для взаимодействия с пользователем.
- 5. Fragment (фрагмент)** – компонент, представляющий поведение или часть пользовательского интерфейса операции.
- 6. Стек** – тип данных, представляющий список элементов, организованный по принципу LIFO.
- 7. LIFO (Last In First Out)** – принцип построения списков, при котором последний добавленный элемент извлекается первым.
- 8. View (представление)** – базовый класс для виджетов, которые используются для создания компонентов интерактивного интерфейса.
- 9. Виджет** – элемент интерфейса.
- 10. Окно, экран** – обособленная часть графического интерфейса, отображаемая пользователю.
- 11. Жизненный цикл** – набор состояний, характеризующий конкретный тип отображения пользовательского интерфейса с момента создания до уничтожения.
- 12. MVP** – шаблон проектирования, основанный на разделении логики на Model – View – Presenter.
- 13. ОС** – операционная система.
- 14. Манифест (AndroidManifest.xml)** – файл, содержащий важную информацию о приложении, которая требуется системе Android.
- 15. Маппинг** – определение соответствия между различными элементами.

16. Status bar – информационная панель в верхней части смартфона, обычно содержащая информацию о текущем времени, заряде батареи, состоянии сети.

17. Router – архитектурный компонент, который должен содержать бизнес-логику, связанную исключительно с переходами между экранов.

18. Deep linking – технология, благодаря которой пользователь может перемещаться между приложениями в заранее определенные разделы.

19. Intent – абстрактное описание выполняемой операции.

20. Microsoft Excel – программа для работы с электронными таблицами.

21. IDE – интегрированная среда разработки.

22. Рефакторинг – контролируемый процесс улучшения кода, без написания новой функциональности.

23. Логирование – форма автоматической записи в хронологическом порядке операций.

24. Профилирование – сбор характеристик работы программы.

25. Нативное приложение – мобильное приложение, разработанное для использования на определённой операционной системе.

26. SDK (Software Development Kit) – набор средств разработки.

27. Support library – набор классов, предоставляющих обратную совместимость версий.

28. Action Bar – панель действий, расположенная в верхней части экрана под элементом Status Bar.

Оглавление

Реферат	10
Список терминов и сокращений	11
Введение	16
Глава 1. Исследование предметной области	18
1.1. Описание предметной области	18
1.1.1. Описание мобильного приложения	18
1.1.2. Описание бизнес-процесса	23
1.2. Обзор существующего решения	28
1.3. Обзор возможных решений	33
1.3.1. Подход с Single Activity	33
1.3.2. Библиотека Cicerone	34
1.3.3. Библиотека Conductor	36
1.4 Вывод по главе	36
Глава 2. Проектирование модуля навигации	38
2.1. UML-моделирование	38
2.2. Функциональное моделирование нового бизнес-процесса	39
2.3. Моделирование потоков данных	41
2.4. Описание бизнес-логики процесса	44
2.5. Вывод по главе	46
Глава 3. Разработка компонентов нового решения	47
3.1. Обоснование выбора средств разработки	47
3.1.1. Обоснование выбора среды разработки	47
3.1.2. Обоснование выбора языка программирования	48
3.2. Описание реализации решения	50

3.3. Использование измененной системы.....	59
3.4. Вывод по главе	61
Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....	62
4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	62
4.1.1. Потенциальные потребители результатов исследования	62
4.1.2. Анализ конкурентных технических решений.....	63
4.1.3. SWOT-анализ.....	65
4.2. Планирование научно-исследовательских работ	67
4.2.1. Структура работ в рамках научного исследования	67
4.2.2. Определение трудоемкости выполнения работ, разработка графика проведения научного исследования	69
4.2.3. Бюджет научно-технического исследования	72
4.2.3.1. Расчет материальных затрат научно-технического исследования	73
4.2.3.2. Расчет затрат на специальное оборудование	73
4.2.3.3. Основная заработная плата исполнителей	74
4.2.3.4. Дополнительная заработная плата исполнителей	76
4.2.3.5. Отчисления во внебюджетные фонды.....	76
4.2.3.6. Накладные расходы	77
4.2.3.7. Формирование бюджета затрат научно-исследовательского проекта.....	77
4.3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	78

4.4. Вывод по главе	79
Глава 5. Социальная ответственность	80
5.1. Правовые и организационные вопросы обеспечения безопасности 80	
5.2. Производственная безопасность	82
5.2.1. Электромагнитные излучения	83
5.2.2. Недостаток естественного освещения	85
5.2.3. Эмоциональные перегрузки.....	86
5.2.4. Опасность поражения электрическим током.....	88
5.3. Экологическая безопасность	89
5.4. Безопасность в чрезвычайных ситуациях	90
5.5. Вывод по главе	91
Заключение	92
Список достижений	93
Список источников	94
Приложение А	97

Введение

Одной из базовых частей любого мобильного приложения является система навигации. Во внешнем проявлении – это то, как система реагирует на действия пользователя. Во внутреннем – набор базовых компонентов, правил и указаний, с помощью которых реализуется новые функциональные возможности открытия окон.

Нетривиальная навигация в худшем случае запутает пользователя, что, как правило, подлежит скорейшей несложной правке. В свою очередь, некорректная ее разработка способна привести не только к невозможности переиспользования и расширения функциональных возможностей, но и к критическим проблемам внутри всего приложения, которые сопровождаются фатальными ошибками, принудительно завершающими работу.

Для заказчика важно иметь гибкий, расширяемый продукт в области навигации. В текущей реализации возникла следующая проблема: добавление новых фрагментов пользовательского интерфейса стало не только неоднозначным и неочевидным, но и иногда требовало как изменения базовых частей, так и вынужденных обходных приемов. Это привело к увеличению длительности разработки новых функциональных возможностей и, как следствие, их стоимости.

Кроме того, от заказчика поступила задача, подразумевающая унификацию с другим клиентским приложением, в котором в области навигации используется подход, несовместимый с текущей реализацией. Для этого требуется модуль с единой «кодовой базой», используемой в обоих случаях.

Таким образом, для решения проблемы необходимо осуществить оптимизацию навигации. Ее проведение возможно либо с разработкой собственного, либо с использованием готового стороннего решения.

Актуальность разработки собственного решения заключается в полном контроле поведения, возможности изменения модуля под нужды заказчика и независимости от сторонних разработчиков.

Целью является проведение оптимизации системы навигации мобильного приложения для увеличения ее гибкости.

Исходя из поставленной цели были выдвинуты следующие задачи:

1. Анализ функциональности текущей версии системы навигации в мобильном приложении.
2. Изучение возможных вариантов для устранения слабых сторон.
3. Реализация нового решения.
4. Выделение навигации в отдельный модуль.

Глава 1. Исследование предметной области

1.1. Описание предметной области

1.1.1. Описание мобильного приложения

В качестве объекта исследования выступает мобильное приложение «Camfrog» корпорации «Camshare» [1]. Это всемирный кроссплатформенный видеочат, являющийся бесплатным под все доступные платформы. Существуют платные подписки, предоставляющие пользователю дополнительные возможности. Присутствует система внутренней валюты, которая покупается за реальные деньги и может быть потрачена на подарки от пользователя к пользователю. Есть возможность вывода полученных подарков в денежном эквиваленте.

На данный момент приложение «Camfrog» под ОС Android насчитывает более 10 миллионов скачиваний и 1 миллиона зарегистрированных пользователей. В режиме реального времени активно работают более 6 тыс. чат-комнат, в которых общаются более 100 тыс. человек.

В приложении предусмотрены следующие потоковые мультимедиа:

1. Peer-to-peer. Общение «один на один» (рис. 1).

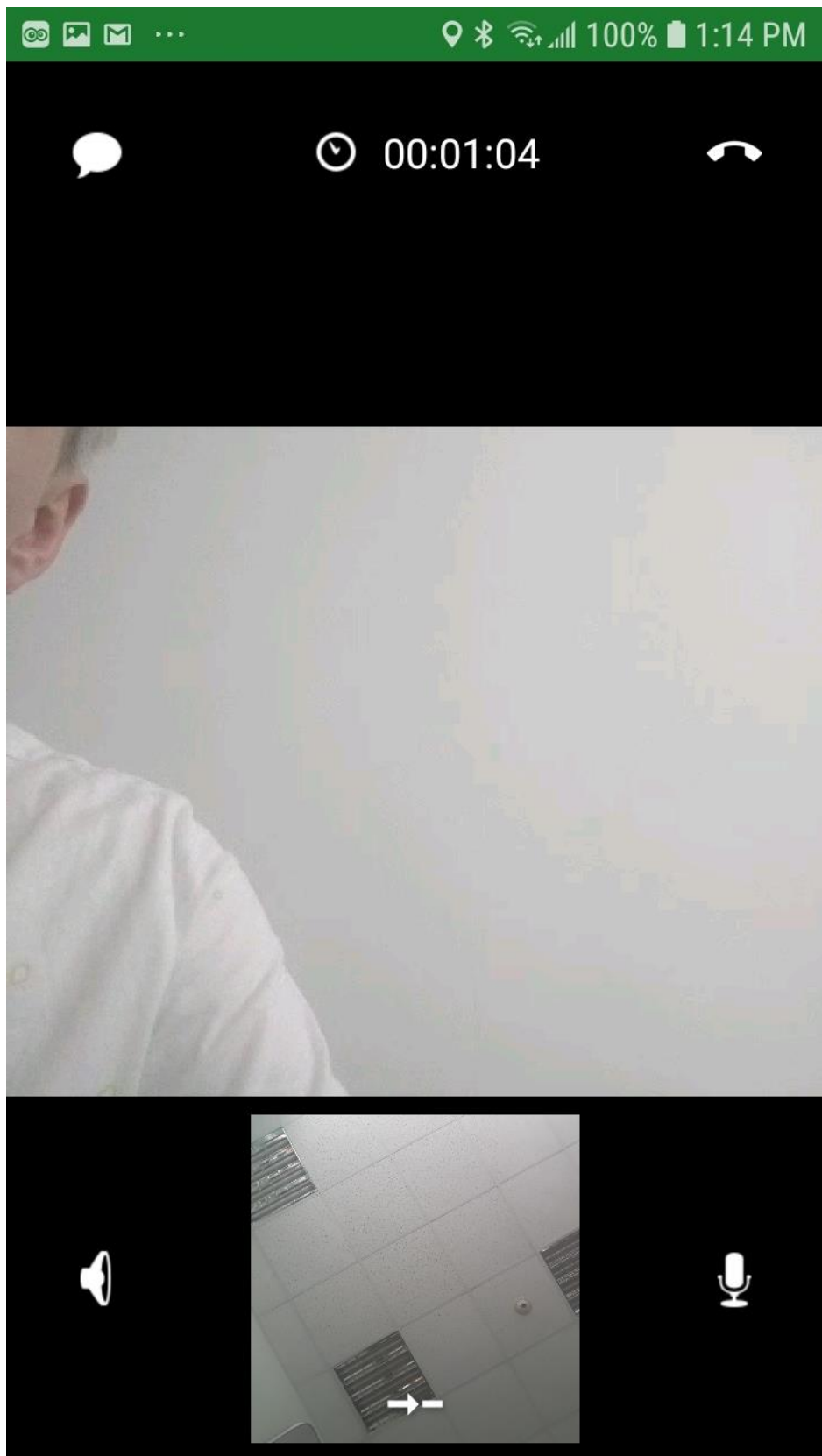


Рисунок 1 – Окно с Peer-to-peer

3. Комнаты. Видеочат с участием большого числа людей, ведущих видеотрансляцию и имеющих возможность просмотра сразу нескольких участников видеочата (рис. 3).

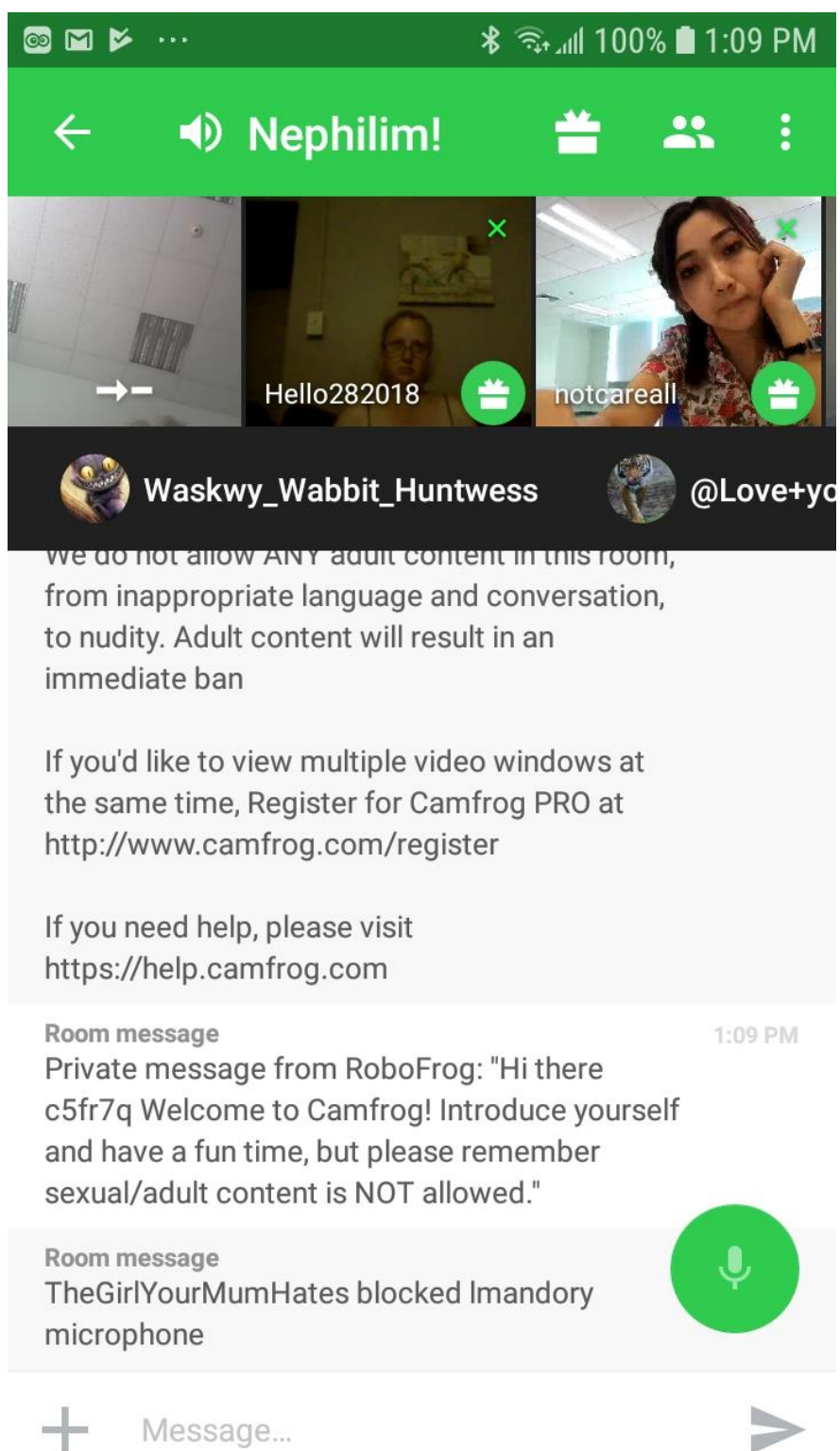


Рисунок 3 – Окно с чат-комнатой

Проект существует более 16 лет и постепенно завоёвывает страну за страной, добавляя новые возможности и совершенствуя существующие. Кроме того, расширяется список поддерживаемых языков. В настоящее время, основная доля пользователей приходится на азиатский регион.

Первое приложение «Camfrog» под ОС Android появилось еще в 2009 году. С тех пор компания Google пересмотрела и переосмыслила весь подход к работе с элементами пользовательского интерфейса, вследствие чего в 2011 году были добавлены новые компоненты [2]. Возвращаясь к проблеме недостаточной гибкости навигации, переход на эти компоненты способствовал ее разрешению, но был невозможен в силу незаконченности данных нововведений, сопровождающихся неконтролируемым поведением и ошибками. Постепенно компания Google совершенствовала операционную систему, исправляя данные ошибки, тем самым побуждая разработчиков к переходу на более современный подход.

В ходе развития приложения заказчиком была выдвинута долгосрочная задача по его объединению с другим продуктом компании – приложением «Paltalk». Данное слияние должно носить внутренний характер с сохранением внешних специфик каждого из клиентов.

Для достижения поставленной задачи лидерами команд Android-разработки было решено вынести базовые компоненты в общий унифицированный модуль. Главной проблемой на пути к решению стало использование принципиально различных компонентов операционной системы, отвечающих за фрагменты пользовательского интерфейса: старых и устоявшихся в «Camfrog» и более новых, но когда-то менее стабильных в «Paltalk». Поскольку навигация базируется именно на них, её унификация стала невозможной.

Исходя из сложившегося конфликта, а также проблемы гибкости существующей навигации в «Camfrog», было решено сначала провести ее оптимизацию внутри данного приложения, а после внедрить и во второе.

1.1.2. Описание бизнес-процесса

Исследуемым бизнес-процессом приложения является добавление логики работы с новым окном. Оно в полной мере характеризует навигацию внутри проекта в целом. Подробное описание данного процесса способствует выявлению слабых сторон, которые привели к проблеме гибкости.

Вначале выполнения задачи на добавление нового окна разработчику предоставляются описание поведения и макеты дизайна, содержащие требуемый заказчиком внешний вид пользовательского интерфейса.

По заданному макету выстраивается разметка в формате XML для ОС Android. Создается пустое окно, для которого применяется подготовленная разметка и реализуется должное поведение, проявляющееся в виде различных обработчиков действий, например, таких как обработка нажатия на кнопку.

После данное окно уже готово к использованию. До полной реализации новой функциональной возможности остается только добавить возможность открыть его. Для этого в компонент, занимающийся исключительно открытием фрагментов пользовательского интерфейса, добавляется соответствующий метод. Далее это окно доступно к переходу из любой точки приложения. В соответствующих местах вызывается данный метод, после чего реализация считается полностью завершённой.

Рисунок 4 графически отображает описанный ранее процесс в виде контекстной диаграммы в нотации IDEF0.



Рисунок 4 – Контекстная диаграмма в нотации IDEF0 бизнес-процесса «Добавление логики работы с новым окном»

В качестве входных параметров процесса выступают:

1. Актуальная версия приложения, которая подлежит модификации.
2. Макеты дизайна, соответствующие представлениям заказчика.

В результате выполнения бизнес-процесса должна быть получена новая работоспособная функциональная возможность, готовая к последующему тестированию и дальнейшему внедрению.

Элементами управления являются:

1. Командное соглашение к стилю программного кода (Code style), соблюдение которого обязательно к исполнению.

2. Функциональные и нефункциональные требования к окну, которое однозначно и, в то же время, полностью раскрывают его. Например, описание того, что должно происходить по нажатию на конкретные элементы графического интерфейса, а также список окон, которые должны находиться в стеке после данного.

3. Возможности базовых компонентов ОС Android, отвечающих за работу с пользовательским интерфейсом. Например, каким образом можно реализовать совместное владение одного виджета в нескольких окнах.

Для того, чтобы произвести данную работу, необходимо задействовать следующие механизмы:

1. Программист, отвечающий за разработку.
2. Персональный компьютер, посредством которого ведется разработка.
3. Интегрированная среда разработки, необходимая для реализации задачи.
4. Средство отображения макета дизайна, позволяющее извлекать различные цвета, картинки, размеры и многое другое.

Далее, на рисунке 5, представлена декомпозиция контекстной диаграммы IDEF0 бизнес-процесса «Добавление логики работы с новым окном».

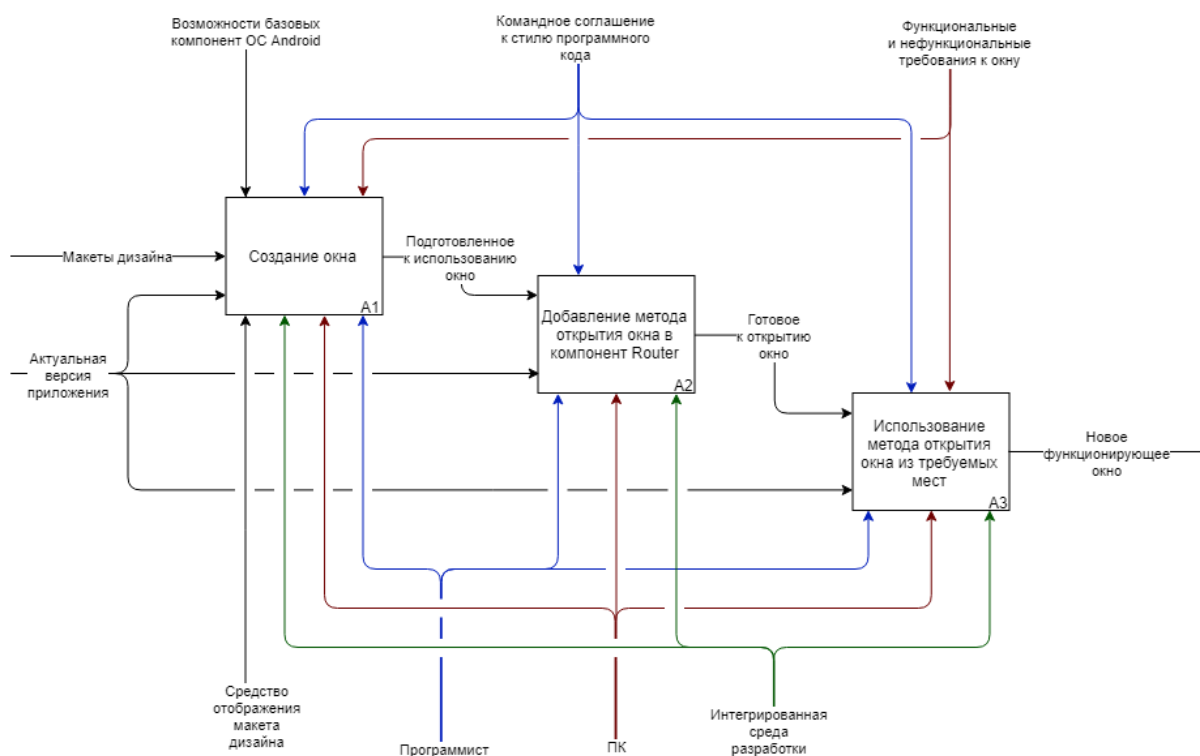


Рисунок 5 – Декомпозиция контекстной диаграммы в нотации IDEF0 бизнес-процесса «Добавление логики работы с новым окном»

Наглядно представлено, что исследуемый бизнес-процесс можно разделить на три подпроцесса, а именно: создание окна, добавление метода его открытия и использование данного метода.

Подпроцесс «Создание окна» может быть декомпозирован (рис. 6).

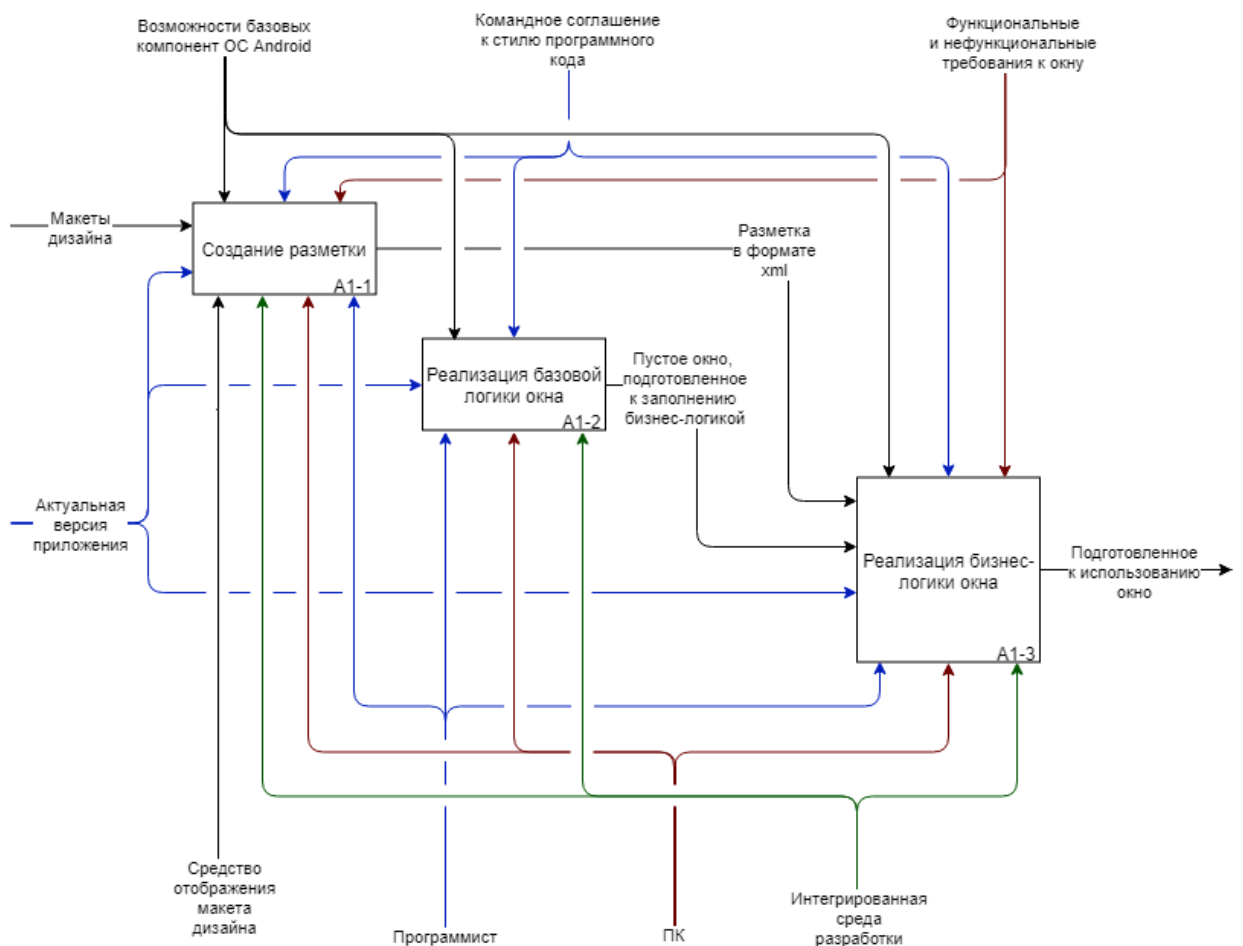


Рисунок 6 – Декомпозиция подпроцесса «Создание окна» в нотации IDEF0

Для создания функционирующего окна необходимо реализовать логику, отвечающую за представление пользовательского интерфейса. Под этим представлением подразумевается набор настроек над окном, характеризующих то, как оно будет показано: например, разрешен ли поворот экрана.

Создается пустой файл с разметкой XML-формата. Далее с помощью специальных средств из предоставленных дизайнерами макетов извлекается необходимая информация, от цветов до картинок под разные размеры экранов, итеративно добавляясь в данную разметку.

На основании пустого окна, описания поведения и разметки реализуется его бизнес-логика. Она включает в себя как привязки событий на элементы пользовательского интерфейса, так и обращения к высокоуровневой, доменной логике приложения, не относящейся ни к базовым функциональным возможностям, ни к навигации.

Для более углубленного поиска причины возникновения проблемы гибкости навигации подпроцесс «Реализация базовой логики окна» был декомпозирован в соответствии с нотацией IDEF0 (рис. 7).

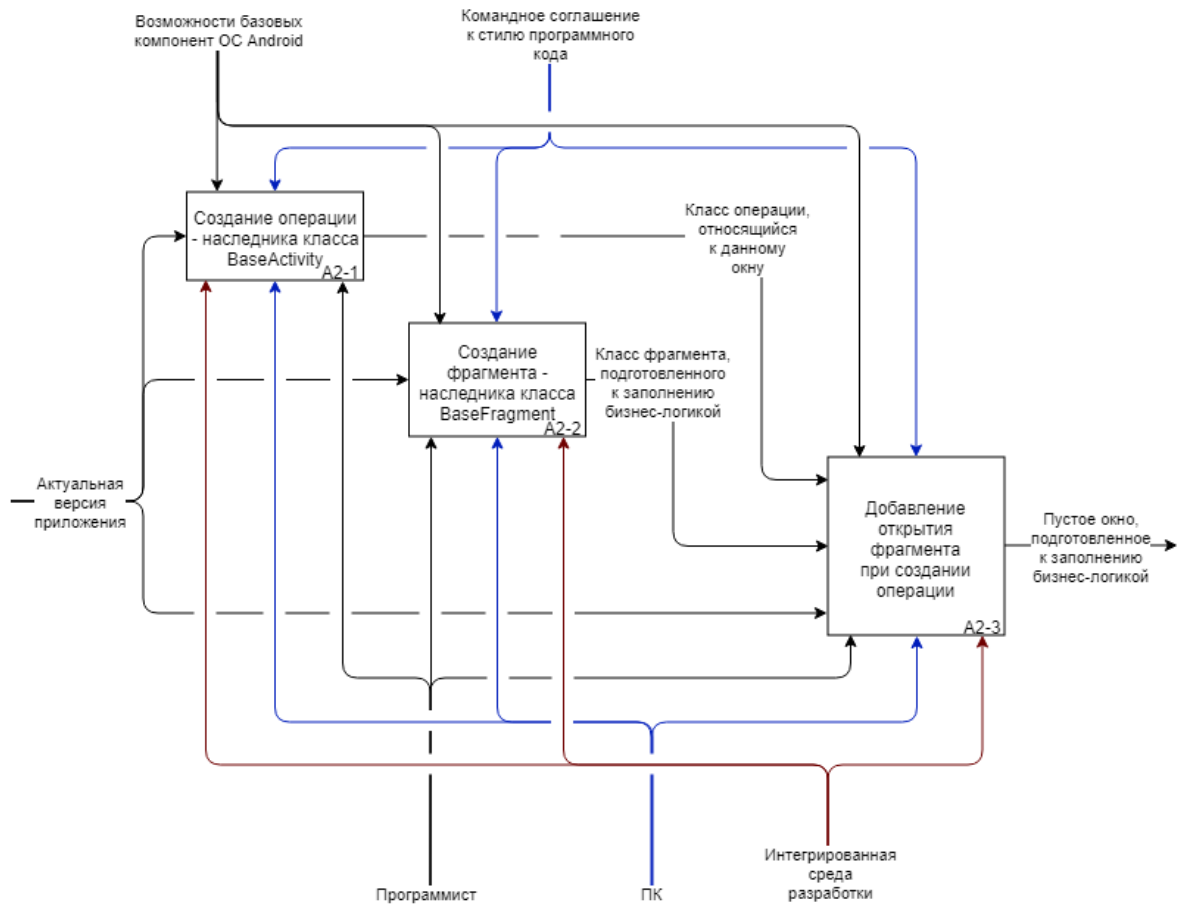


Рисунок 7 – Декомпозиция подпроцесса «Реализация базовой логики окна» в нотации IDEF0

Можно заметить, что для реализации базовой логики одного окна используется два различных компонента, отвечающих за работу с отображением пользовательского интерфейса. Причем первый, тяжеловесный (операция) является лишь контейнером для второго (фрагмента). В силу того, что операция существует с момента появления Android, в ней содержится множество устаревших, неиспользуемых функциональных возможностей, которые накладывают свои ограничения. Во многих моментах другой компонент работы с отображением пользовательского интерфейса, фрагмент, лишен данных ограничений.

1.2. Обзор существующего решения

Для того, чтобы понять, как решить проблему недостаточной гибкости, присутствующую в текущей реализации, необходимо выявить характер её проявления, существующие слабые стороны.

При построении навигации на подходе с использованием операций между окнами создается стек экранов. Данный стек закрыт к манипуляциям для разработчика. Это делает труднодоступным, иногда и вовсе невозможным контроль переходов между существующими окнами.

В ходе добавления новых функциональных возможностей в области навигации нередко возникает потребность в добавлении сразу нескольких окон в стек. Видимым экраном должен быть тот, который был добавлен последним. По переходу назад, осуществляемого с помощью системной кнопки «Back», должен открываться не тот экран, с которого текущий был запущен, а предшествующий данному в стеке. Из-за отсутствия контроля над этим элементом сложно добиться требуемого результата.

В рамках текущего подхода существуют следующие способы решения данной проблемы:

1. Заблаговременно открывать окна последовательно, до нужного. В таком случае будут видны все переходы от одного окна к другому до заданного, что совершенно не нужно пользователю, может вызвать его недовольство и, как следствие, оставить не самые лучшие впечатления от приложения. Для человека, использующего какое-либо приложение, даже не обязательно мобильное, различные переходы должны выглядеть просто, красиво и интуитивно понятно, без чего-то лишнего.

2. Переопределить стандартное поведение перехода назад, вместо этого открывая новое окно, которое должно быть предшествующим в стеке. Такое решение крайне нежелательно, так как требует дальнейшего повсеместного использования. В данном случае разработчик «ломает» внутреннюю реализацию и навязывает свою. Это не только требует дополнительного тестирования и сопровождения, но и ведет к захламлению

архитектуры приложения в области навигации. Кроме того, такая реализация имеет дополнительные сложности, связанные с тем, что текущее окно «знает» о том, какое конкретно ему предшествует. Различные предшествующие элементы будут требовать дополнительной реализации.

Реальным примером является процесс открытия трансляции или чат-комнаты. В соответствии с требованиями, когда пользователь осуществляет сначала переход на данное окно, а после – назад, перед ним должен открыться экран со списком всех медиа соответствующего типа.

Внешне похожая, но в то же время иная проблема существует из-за варьирующейся точки входа в приложение. В ОС Android каждое уведомление относится к какой-то конкретной операции, которая и открывается по нажатию. Когда пользователь разворачивает уведомление о том, что другой пользователь прислал ему сообщение, должно открыться окно чата, а по переходу назад из него – либо предыдущий экран, либо, в случае отсутствия последнего, главный. Стоит заметить, что при открытии главного окна происходит системная очистка стека, которая возможна при помощи специальных флагов. В рамках подхода с применением операций данное поведение не достижимо в требуемой форме. Любая реализация приведет лишь к частичному результату: либо будет открываться предыдущий экран, сопровождаемый закрытием приложения в случае его отсутствия, либо последовательно, главное окно, а после чат, что делает невозможным возврат на предшествующее.

К проблеме стека также относится отсутствие простой проверки того, было ли конкретное окно открыто или нет. И хотя в текущем приложении необходимость в этой функциональной возможности отсутствует, потенциально она все же может когда-то потребоваться.

Еще одна проблема гибкости навигации связана с различием пользовательского интерфейса в зависимости от типа устройства. На телефоне существуют два отдельных окна: список всех чатов и чат с конкретным пользователем; на планшете же они объединены в одно (рис. 8, 9).

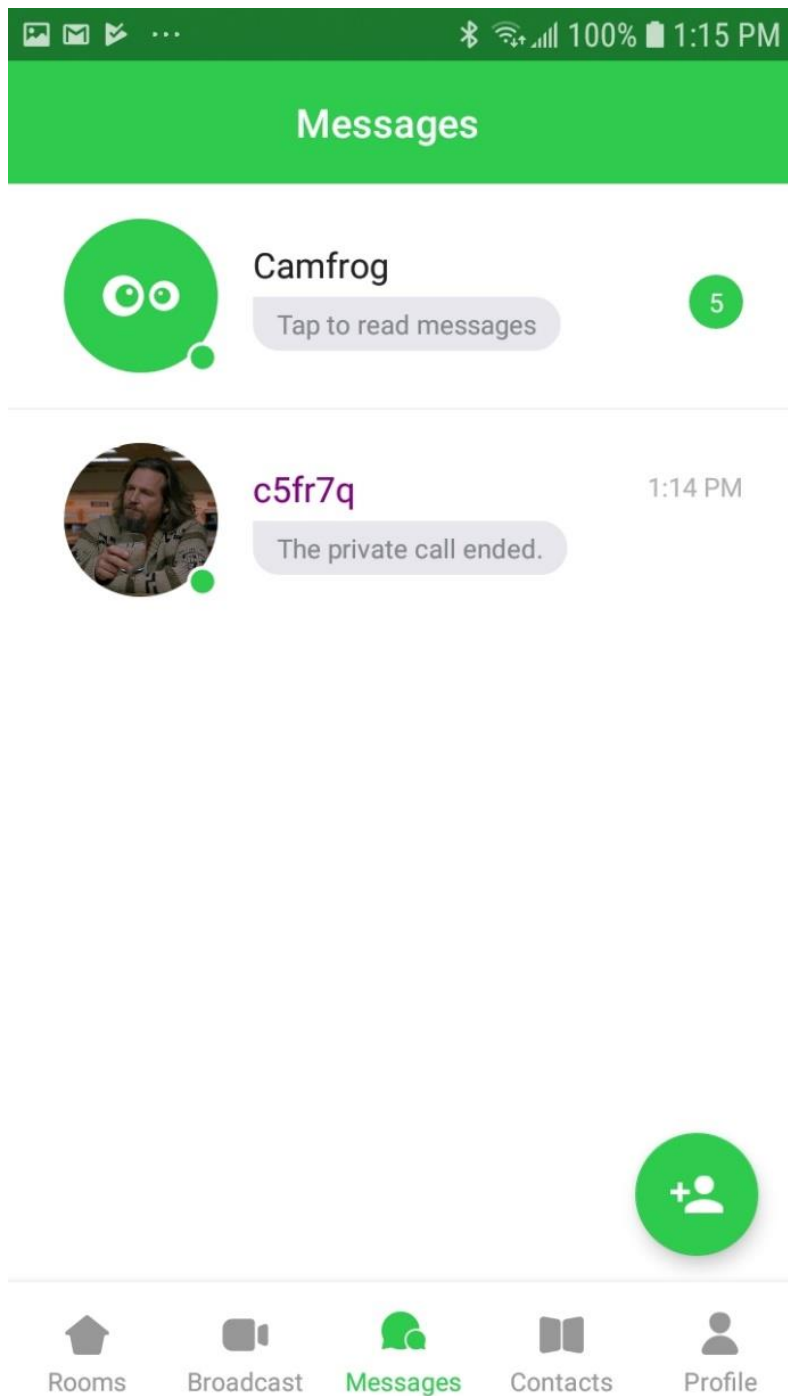


Рисунок 8 – Окно со списком всех чатов на телефоне

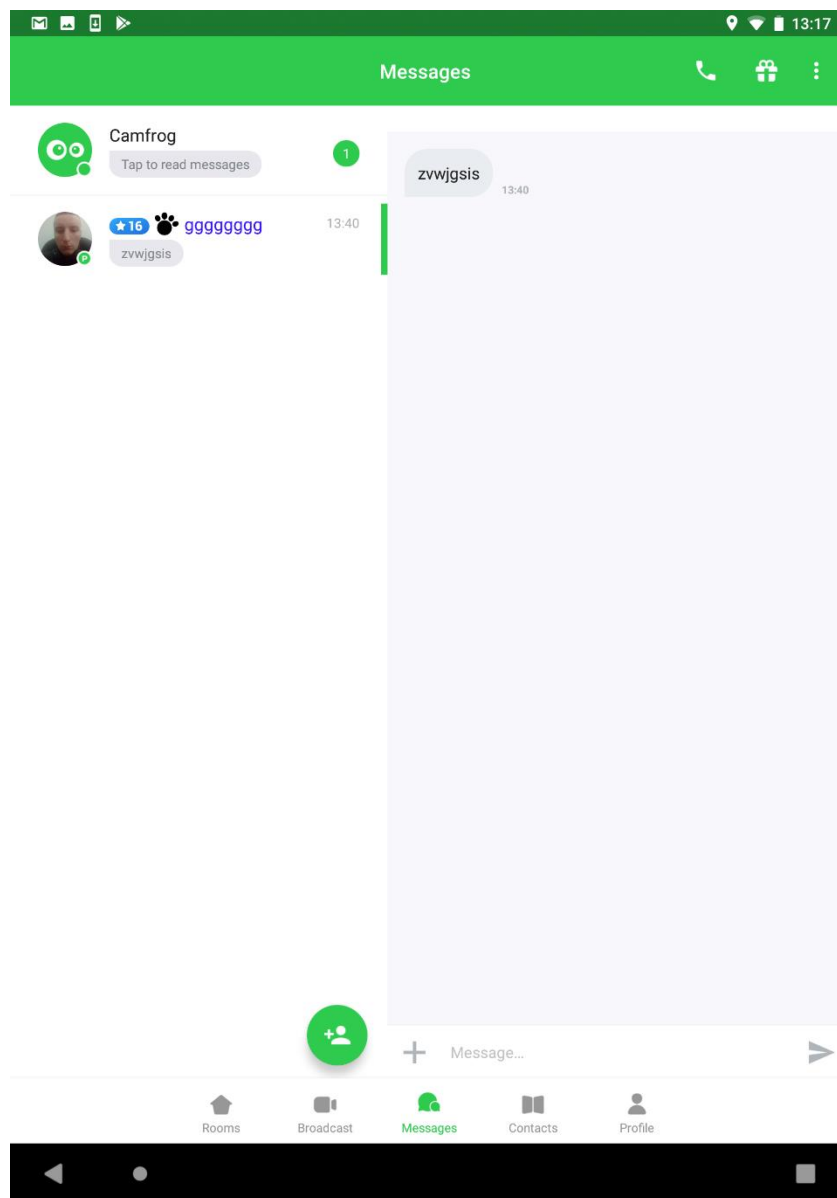


Рисунок 9 – Окно со списком всех чатов и чатом с конкретным пользователем на телефоне

С учетом этого, для двух логических экранов существует три физических (в виде операций): по одному для каждого варианта и один отдельный для совмещенного. Поэтому, например, возникает вынужденное дублирование программного кода, которое следует сопровождать и в дальнейшем: при добавлении нового типа уведомления, ведущего конкретно к одному из указанных окон, разработчику следует также дополнить совмещенный вариант абсолютно аналогичным кодом.

Подход с использованием операций также не дает полной свободы творчества в области визуализации переходов между окнами. С момента

создания системы существует возможность применения элементарных анимаций, таких как замена экранов с помощью плавного сдвига или изменения прозрачности. Для них можно настраивать длительность, начальные и конечные значения (положение в пространстве и прозрачность), а также направление для сдвига. В 2014 году компания Google презентовала Android 5.0 Lollipop [2]. Одним из нововведений стала возможность анимировать переход между общими для операций элементами графического интерфейса. По умолчанию параметры рассчитываются автоматически, но существует возможность частичной настройки: с указанием длительности или типа траектории (прямолинейной или криволинейной). Тотальный контроль разработчиком анимаций перехода в рамках текущего подхода невозможен по сей день.

Кроме всех негативных сторон, описанных выше, в существующем решении есть ряд положительных черт:

1. Долгий срок существования компонента – операции, его использование и официальное принятие компанией – основателем данной операционной системы, делает его самым стандартным и наиболее применимым. Следствием этого является огромное сообщество, способствующее разрешению спорных ситуаций (в частности, с навигацией).

2. Наличие возможностей настройки флагов, отвечающих за поведение экрана. Среди них: возможность поднятия, сжатия контента или бездействия в качестве реакции на открытую клавиатуру; возможность назначения доступной ориентации экрана: только портретной, только ландшафтной или любой из них, а также некоторые другие.

Декомпозиция слабых и сильных сторон существующего решения применительно к навигации графически представлена в виде диаграммы Исикавы на рисунке 10.

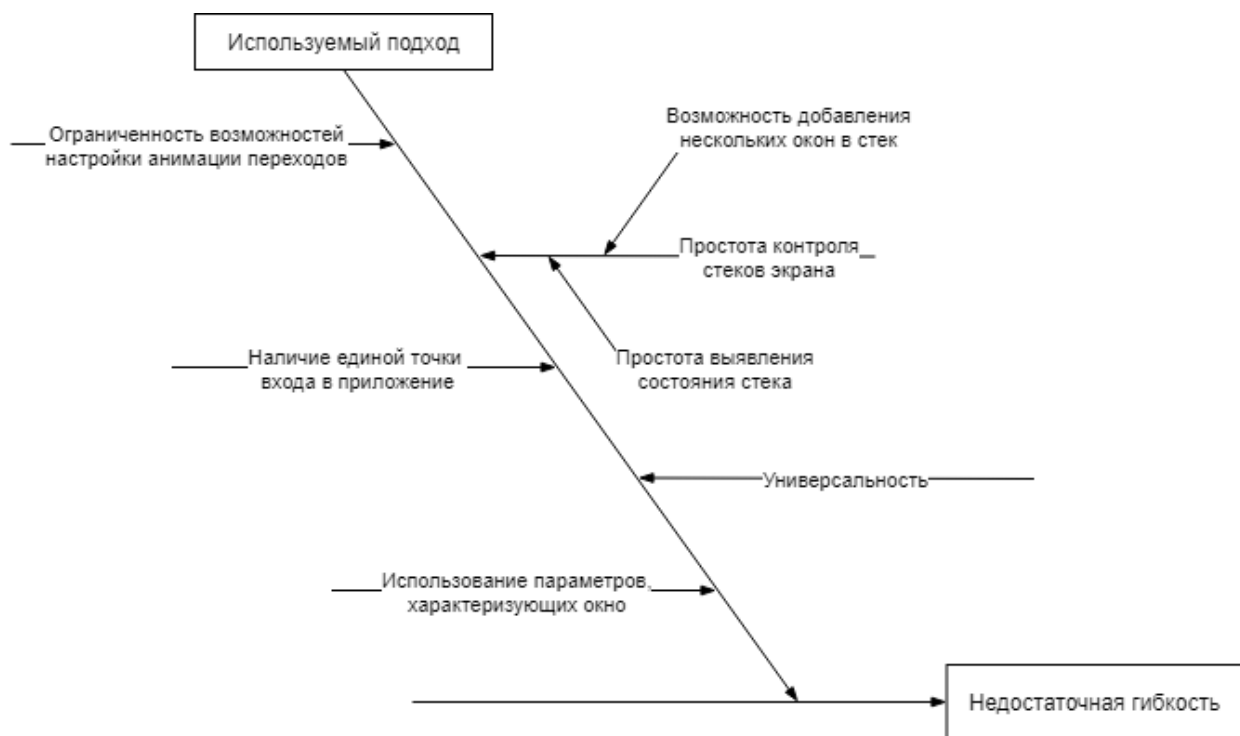


Рисунок 10 – Причинно-следственная диаграмма появления проблемы недостаточной гибкости модуля навигации

1.3. Обзор возможных решений

1.3.1. Подход с Single Activity

Наиболее популярное решение проблемы гибкости модуля навигации – реализация подхода, при котором существует лишь одна операция, являющаяся точкой входа в приложение и контейнером всех экранов. Сами окна при этом характеризуются наследником другого компонента работы с отображением пользовательского интерфейса ОС Android, фрагментом.

Популярность данного решения связана с его сильными сторонами:

1. Предоставляется единая точка входа в приложение.
2. Контроль стека экранов становится простым как к пониманию, так и к применению.
3. Используемые компоненты общеприменимы и считаются стандартными также и вне подхода.
4. Доступна возможность отрисовывать анимации перехода вручную [3].

5. Основной компонент подхода – фрагмент – не потребляет большого объема системных ресурсов.

6. Отсутствует работа с устаревшими, на деле неиспользуемыми компонентами.

7. Жизненный цикл приложения становится наглядным для разработчика.

Также можно выделить следующие недостатки:

1. Некоторое поведение, ранее управляемое с помощью флагов (такое как полноэкранность и реакция на клавиатуру), теперь необходимо контролировать вручную.

2. Выявления требований к базовому компоненту, характеризующему окно, для больших проектов является достаточно непростой задачей в силу своего объема.

3. Присутствует сложность в реализации удовлетворяющих выявленным требованиям функциональных возможностей, связанная с анализом данного требования на всех окнах.

1.3.2. Библиотека Cicerone

«Cicerone» – это библиотека, оказывающая малое влияние на производительность, которая облегчает работу с навигацией в Android-приложении. Разработана с расчетом на MVP, но также может быть задействована с использованием любого другого архитектурного паттерна [4].

На рисунке 11 представлена структура работы библиотеки, посредством которой происходят переходы между окнами.

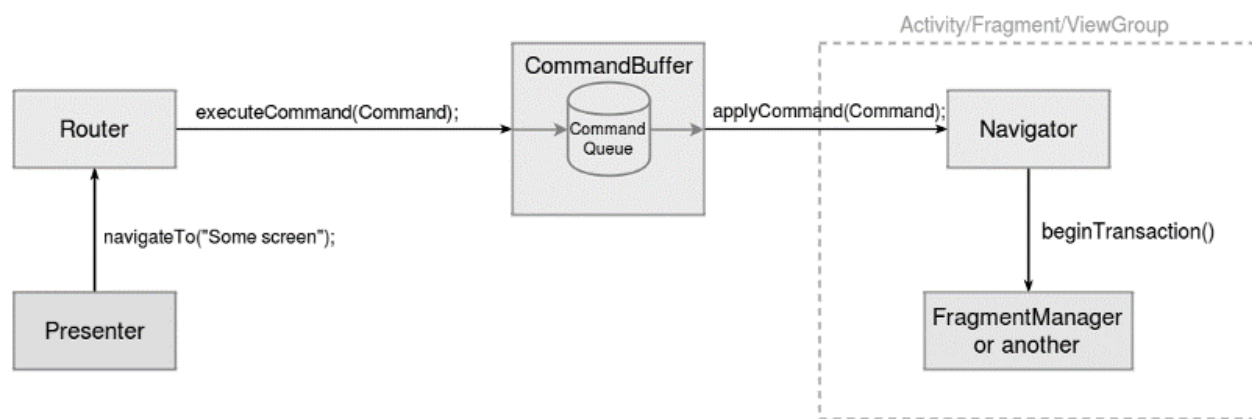


Рисунок 11 – Структура библиотеки Cicerone

Реализация навигации построена с использованием поведенческого паттерна «Команда». Существуют следующие типы команд, регламентирующие поведение экрана в стеке: для добавления – `forward`; для замены – `replace`; для перехода назад – `back`; для перехода назад к конкретному окну – `backTo`.

В библиотеке нет привязки к конкретному компоненту Android для отображения пользовательского интерфейса, что делает возможным переходы между различными типами реализаций окна. От разработчика требуется реализация интерфейса `Navigator`. Существует уже готовая реализация с использованием компонента-фрагмента.

Достоинствами «Cicerone» являются:

1. Отсутствие привязки к конкретному компоненту отображения пользовательского интерфейса.
2. Предоставление возможности коротких вызовов.
3. Легкость в расширении.

У библиотеки существуют следующие недостатки:

1. На практике четырех команд может быть недостаточно.
2. Отсутствие привязки к конкретному подходу позволяет построить навигацию с использованием различных базовых компонентов. Это лишает положительных сторон какого-либо из подходов, а также вносит различные ограничения.

1.3.3. Библиотека Conductor

«Conductor» позиционируется как замена стандартным компонентам-фрагментам. Основная идея – «обернуть» класс представления и дать доступ к методам жизненного цикла. «Conductor» имеет свой жизненный цикл, упрощенный по сравнению с аналогами из операциями и фрагментами [5].

Положительными сторонами библиотеки являются:

1. Построение приложения на одной операции.
2. Отсутствие привязки к архитектуре.
3. Возможность легко встроить анимацию переходов.
4. Доступность работы с самописным стекком экранов.
5. Наличие различных готовых анимаций.
6. Наличие плавности анимаций переходов между окнами любой сложности.

Кроме этого, данное решение сопровождается рядом серьезных проблем:

1. Использование нестандартных компонентов, изначально не предназначенных для навигации.
2. Постоянно пополняющийся широкий список задач, связанных с фатальными ошибками, сопровождающимися закрытием приложения.

1.4 Вывод по главе

В результате обзора для каждой альтернативы существующей реализации навигации были выявлены сильные и слабые стороны. Кроме собственных недостатков готовые решения отрицательно воспринимаются в рамках команды, так как подразумевают под собой зависимость от создателей библиотек и, как следствие, уменьшение подконтрольности разработки. Наличие фатальных ошибок в таких решениях вовсе недопустимо, так как последние не могут быть исправлены, вынуждая оставить приложение с неисправностью в ожидании обновлений библиотеки. Из-за этого вариант с применением библиотеки «Conductor» был отклонен.

Применение «Cicerone» было возможным, но конечное решение лидерами команд было сделано в сторону собственной реализации подхода с Single Activity, что стало свидетельством актуальности разработки.

На данное решение повлияли следующие аспекты:

1. Необходимость сохранения независимости базовой части приложения от воздействий сторонних библиотек.
2. Применение подхода с Single Activity одним из приложений.
3. Решение наиболее важных, поведенческих проблем, существующих в текущей реализации «Camfrog».

Глава 2. Проектирование модуля навигации

2.1. UML-моделирование

В работе рассматривается система навигации с точки зрения её реализации, а не взаимодействия с ней пользователя. В рамках этого предусматривается всего одна роль: *разработчик*.

Для определения функциональных возможностей системы для данной роли была составлена диаграмма вариантов использования в нотации Unified Modeling Language (UML). Она описывает взаимоотношения и зависимости между группами вариантов использования и действующих лиц, участвующими в процессе [6]. Каждая возможность определяется вариантом использования. Данная диаграмма представлена на рисунке 12.

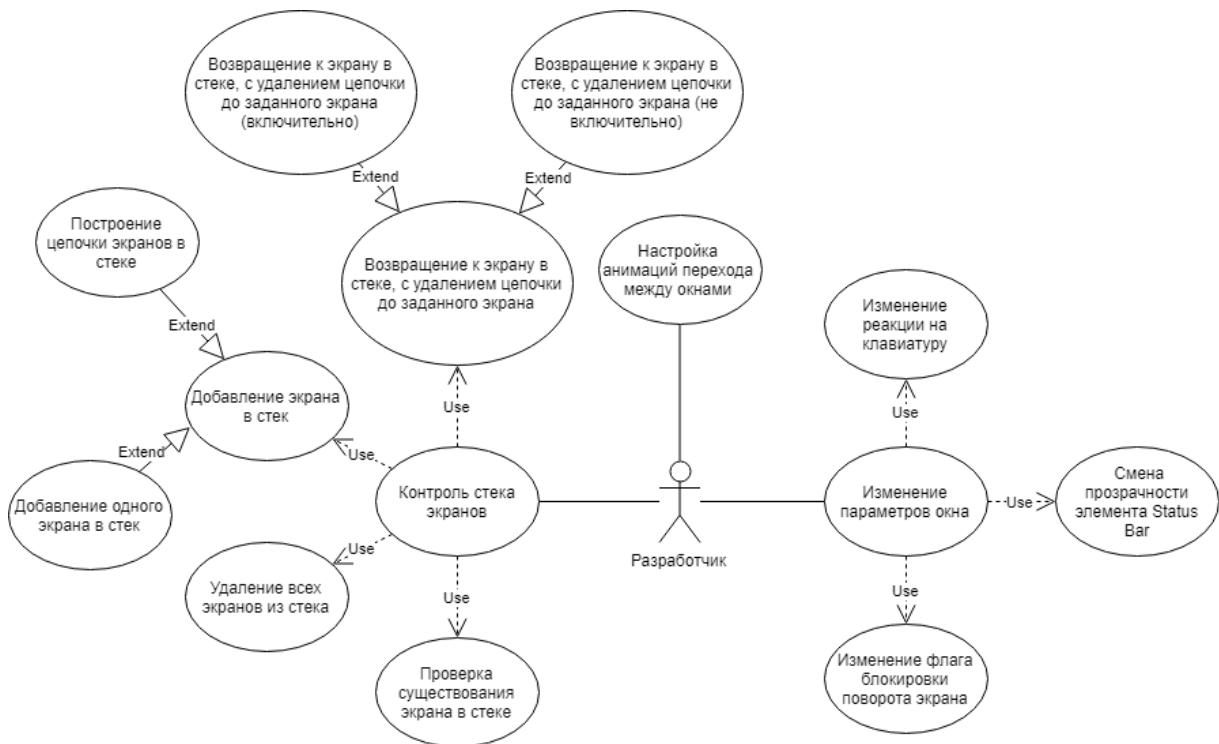


Рисунок 12 – Диаграмма вариантов использования системы навигации в нотации UML

Наглядно показано, что после проведения оптимизации *разработчику* должны быть доступны три следующие группы возможностей:

1. Работа со стеком экранов.
2. Изменение параметров окна.
3. Настройка анимаций переходов.

Данные группы расположены в следующем порядке: от более значимой, до менее значимой.

2.2. Функциональное моделирование нового бизнес-процесса

Для достижения гибкости системы навигации была поставлена задача оптимизации данной системы. Способом ее решения является пересмотр существующего бизнес-процесса и изменения одного или нескольких содержащихся подпроцессов путем реализации подхода с Single Activity.

На этапе описания бизнес-процесса было выявлено, что проблема содержится в подпроцессе «Реализация базовой логики окна», описанном ранее и представленном на рисунке 7. Детальный пересмотр имеющегося решения показал необходимость модернизации подпроцесса более высокого уровня «Создание окна».

Данные изменения представлены на рисунке 13 в виде диаграммы в нотации IDEF3. Эта методология предназначена для описания процессов, выполняющихся в определенной последовательности, а также объектов, участвующих совместно в одном процессе [7].

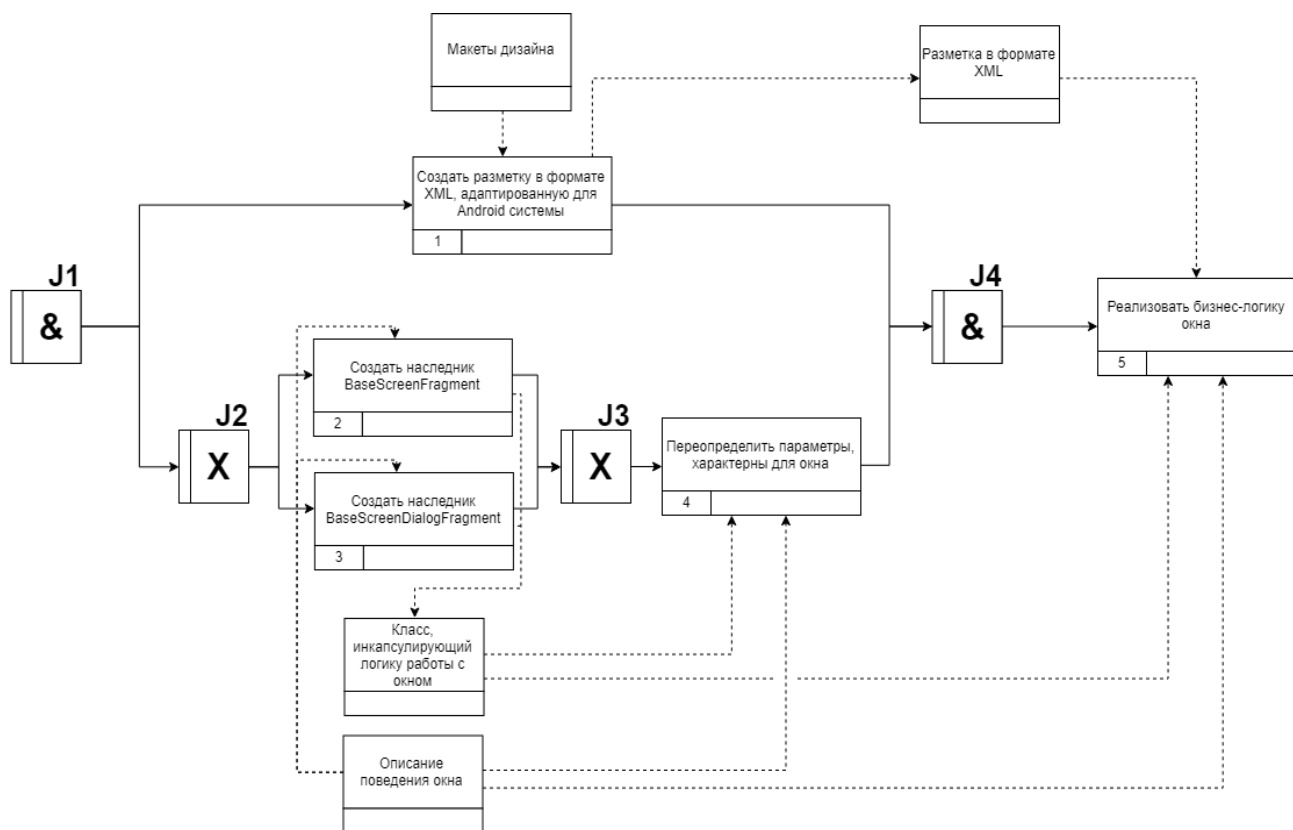


Рисунок 13 – Диаграмма в нотации IDEF3 для подпроцесса «Создание окна»

В диаграмме демонстрируется применение подхода с Single Activity, при котором для добавления окна не создается новая операция. Характеризуется это отсутствием использования BaseActivity, а также дроблением класса фрагмента, BaseFragment, на два более узконаправленных: BaseScreenFragment, BaseScreenDialogFragment. Кроме того, для нового подпроцесса присуще переопределение характерных для окна параметров сразу после создания компонента, ранее являющихся частью реализации бизнес-логики.

Данные изменения возникли в силу имеющегося недостатка в новом подходе, подразумевающего ручное управление поведением прежде контролируемого с помощью флагов. Так, раньше эти флаги выносились в отдельный файл манифеста и координировались на уровне ОС. Среди них была возможность по-разному показывать операцию: в виде диалогового окна или во весь размер экрана телефона. При использовании фрагментов требуется применение наследования от разных базовых компонентов ОС Android.

2.3. Моделирование потоков данных

Посредством взаимодействия разработчика с модулем навигации происходит создание новой функциональности, представляемой для пользователя в виде окон.

Для описания того, каким образом происходит отображение реализованных окон, был составлен ряд диаграмм потоков данных с применением методологии Data Flow Diagrams (DFD). Эта нотация предназначена для моделирования информационных систем с точки зрения хранения, обработки и передачи данных [8].

На рисунке 14 представлена общая диаграмма потоков данных того, как отображаются окна пользовательского интерфейса.



Рисунок 14 – Диаграмма потоков данных в нотации DFD для процесса «Отображение окон пользовательского интерфейса»

Графически представлено, что вследствие взаимодействия пользователя с мобильным приложением происходит отображение контента, представленного в виде окон.

Мобильное приложение – это сложная система, состоящая из множества составляющих. На рисунке 15 представлена диаграмма потоков данных с учетом декомпозиции приложения, применительно к процессу отображения окон.



Рисунок 15 – Диаграмма потоков данных в нотации DFD для процесса «Отображение окон пользовательского интерфейса» с декомпозицией сущности «Мобильное приложение»

Данная диаграмма демонстрирует то, как система навигации участвует в отображении окон. В мобильном приложении (в целом) пользователь взаимодействует исключительно с пользовательским интерфейсом. Тот, в свою очередь, отслеживает изменения собственного состояния, а после, посредством передачи действия, оповещает модуль навигации об этом изменении.

Действия, протекающие между сущностями пользователь-интерфейс и интерфейс-навигация, имеют различное представление. Для первого случая это различные нажатия на кнопки, разворачивания уведомлений приложения, переход из сторонних источников по ссылкам, ведущих в данное приложение.

Для второго случая все действия в области переходов делятся на две группы:

1. Явные, когда пользователь взаимодействует с текущим приложением. Например, нажимает на кнопку «Сообщения» и ожидает переход на соответствующее окно со списком всех сообщений. В таком случае действие – это запрос конкретного окна для отображения.

2. Неявные, когда пользователь взаимодействует со сторонними источниками или системой в ожидании открытия приложения. Например, разворачивает уведомление или переходит по ссылке, ведущей в данное приложение. В этом случае действие – это запрос окна для отображения, тип и параметры которого сначала нужно определить.

С учетом этого поведение внутри системы навигации будет варьироваться. Данное различие представлено в виде диаграммы потока данных на рисунке 16.

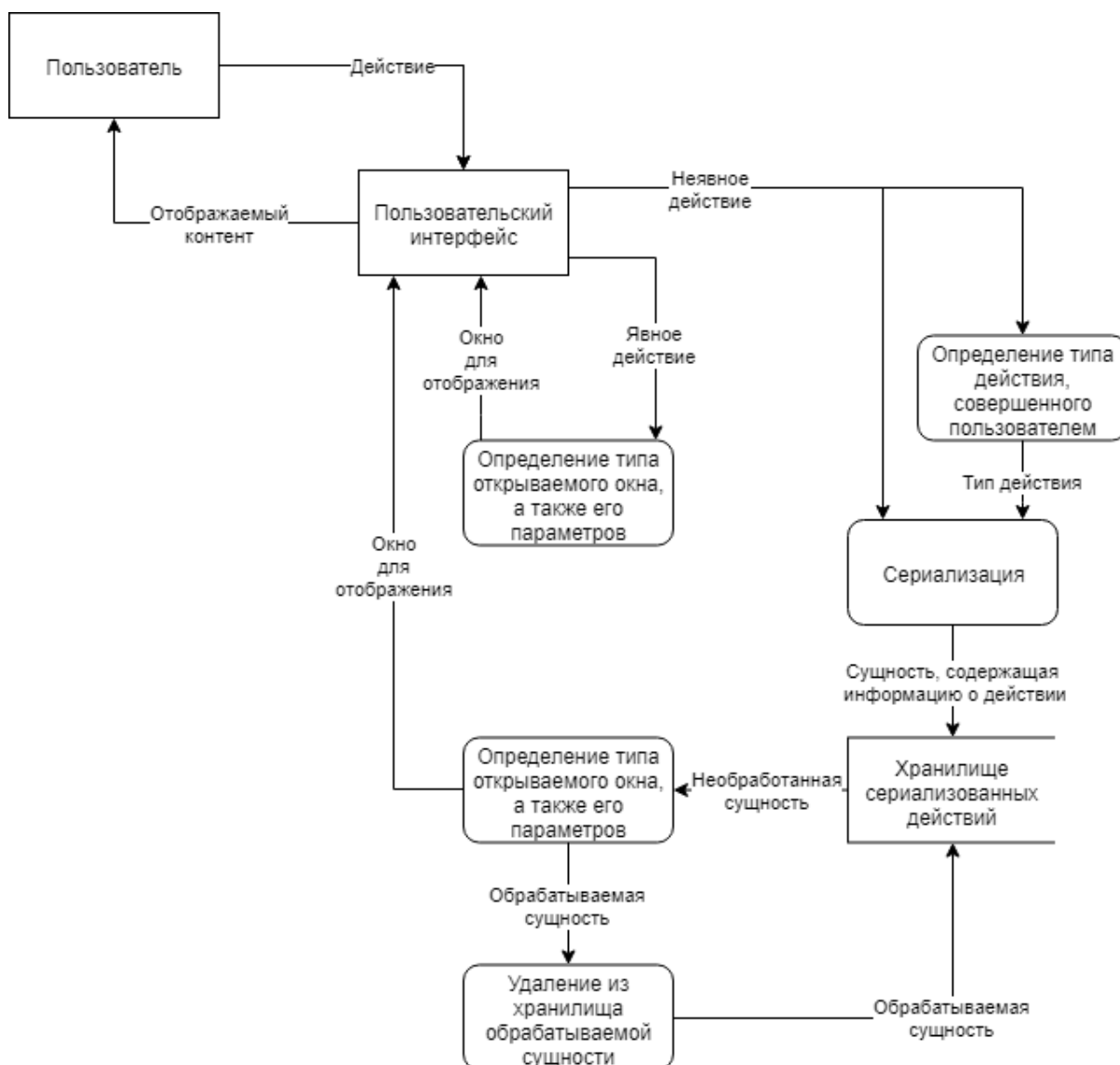


Рисунок 16 – Диаграмма потоков данных в нотации DFD для процесса «Отображение окон пользовательского интерфейса»

Диаграмма показывает, что за неявным действием следует более сложный процесс обработки. Для этого случая также характерно использование хранилища действий (с сохранением в область памяти приложения). Это связано с необходимостью выдачи окна для отображения только после того, как приложение инициализировано и соединение с сервером установлено.

2.4. Описание бизнес-логики процесса

Для описания бизнес-логики системы навигации после внесения изменений был построен ряд предназначенных для этого диаграмм.

На рисунке 17 представлена диаграмма в нотации Business Process Model and Notation (BPMN) применительно к оптимизированному модулю навигации. Данная методология используется для представления алгоритма выполнения процесса [9] и применяется для создания процессно-ориентированных систем, где человек играет подчиненную, а система – ведущую роль [10].

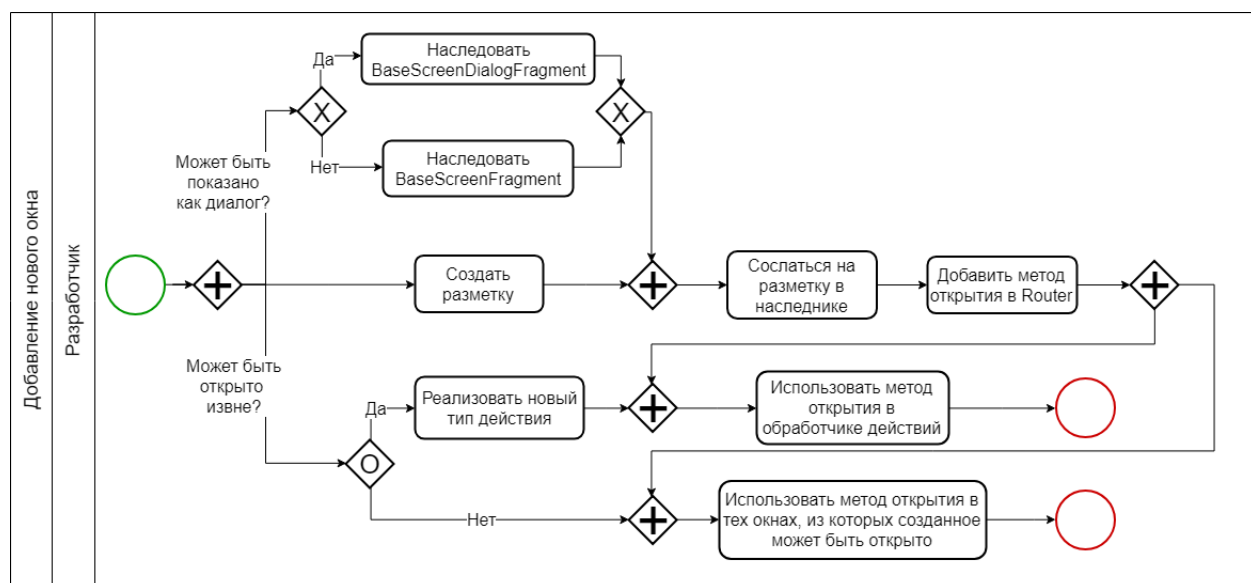


Рисунок 17 – Диаграмма моделирования бизнес-процесса «Добавление нового окна» в нотации BPMN

Диаграмма кратко описывает, что должен делать программист для создания нового окна. Наглядно представлено, что алгоритм реализации варьируется в зависимости от следующих параметров:

1. Возможность окна быть показанным в качестве диалогового.
2. Возможность открытия данного окна извне.

Более подробное описание алгоритма процесса добавления логики нового окна представлено в диаграмме моделирования бизнес-процессов в нотации Event-driven Process Chain (EPC) на рисунке 18. Нотация моделирования EPC ориентирована на построение алгоритмов взаимодействия в процессе выполнения конкретной работы [11].

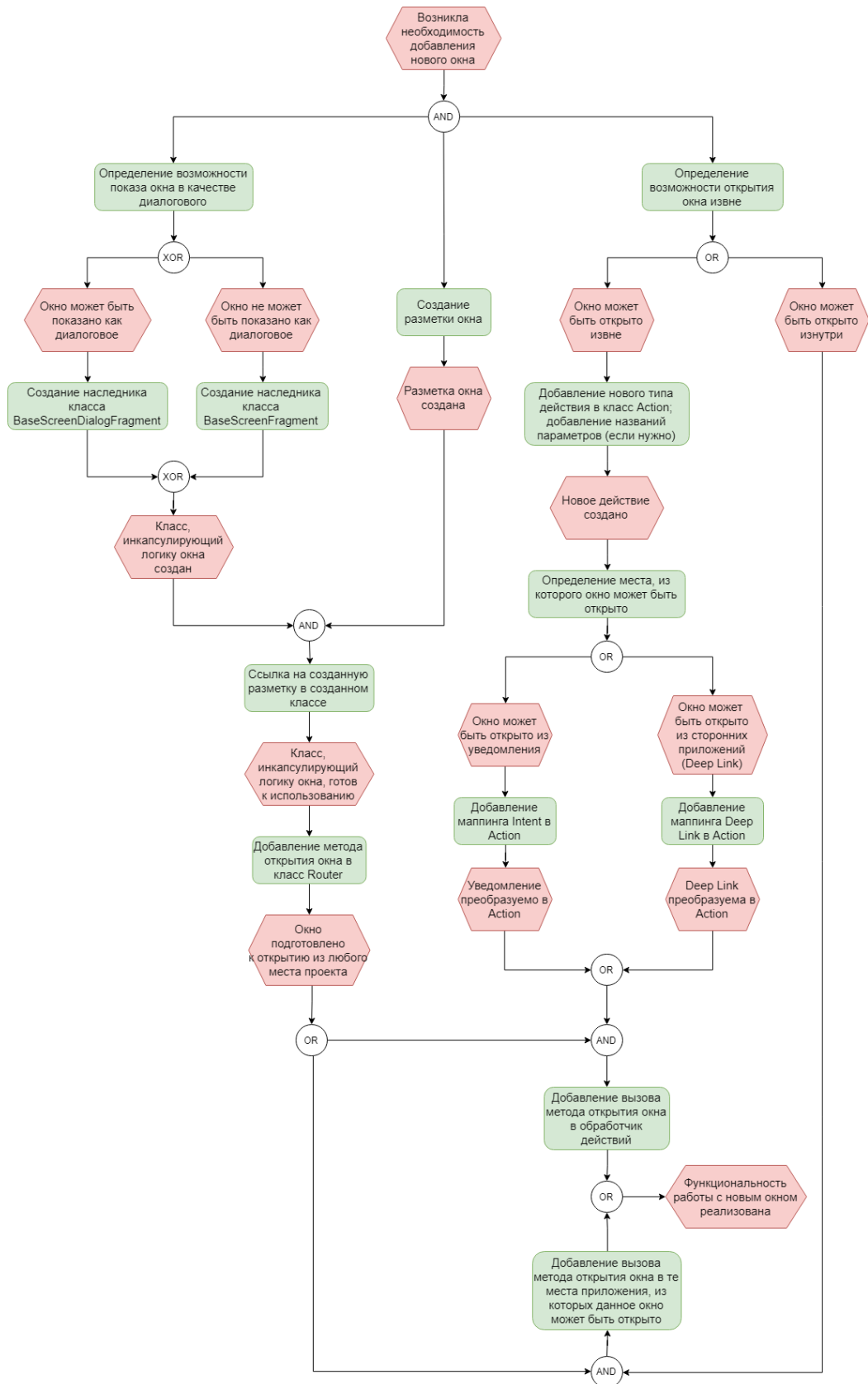


Рисунок 18 – Диаграмма моделирования бизнес-процесса «Добавление нового окна» в нотации EPC

Данная диаграмма более подробно описывает алгоритм для случая неявного пользовательского действия в области навигации.

Каждое неявное действие имеет различного рода место происхождения. В случае с уведомлениями приложения, в систему посылается специфичный для Android ОС компонент с различными параметрами – Intent. В других случаях посылается Intent со строго одним строковым параметром-ссылкой, преобразуемый в класс Deep Link.

В оптимизированной системе любой из посылаемых компонентов преобразуется в компонент навигации Action, характеризующийся типом и списком параметров. Последний может быть пустым, в зависимости от конкретного окна. Каждый параметр характеризуется названием и значением. Примерами преобразования неявных действий в Action без учета промежуточных компонентов являются:

1. Неявное действие для открытия окна профиля пользователя, состоящее из типа «Профиль» и пустого списка параметров.

2. Неявное действие для открытия окна трансляции, характеризующееся типом «Трансляция» и списком параметров, содержащих один из параметров: либо «уникальный идентификатор пользователя» с числовым значением, либо «имя пользователя» в системе «Camfrog» со строковым значением.

2.5. Вывод по главе

В рамках данной главы было проведено проектирование модуля навигации с использованием ряда диаграмм. Представлены способы по устранению слабых сторон, основные прецеденты использования системы, а также особенности последующей реализации.

Глава 3. Разработка компонентов нового решения

3.1. Обоснование выбора средств разработки

3.1.1. Обоснование выбора среды разработки

Выбирая среду разработки под ОС Android, почти каждый программист остановится на Android Studio, однако перечень возможных средств не состоит из одного. Хотя итоговое решение осталось за данной средой, необходимо рассмотреть все возможные решения.

IDE Eclipse – это бесплатная среда разработки от некоммерческой организации Eclipse Foundation [12]. Сама программа является некоторой основой, к которой подключаются различные модули, позволяя разрабатывать под разные языки программирования.

Она обладает следующими преимуществами:

1. Наличие русскоязычного интерфейса и документации.
2. Отличная производительность на «слабых» машинах.
3. Большое число доступных расширений.
4. Возможность подключения модулей.

Наиболее популярна среда была до 2014 года, когда считалась монополистом на рынке IDE для данной операционной системы, а после компания Google представила Android Studio, провозгласив ее как основную среду для разработки приложений под Android.

IntelliJ IDEA – это среда разработки, созданная отечественной компанией JetBrains [12]. Как и Eclipse, эта среда разработки даёт возможность создавать программы на нескольких языках программирования. По сравнению с конкурентом имеет ряд расширенных возможностей.

Среди преимуществ выделяются:

1. Быстрая отладка значений.
2. Автозаполнение методов.
3. Наличие возможности рефакторинга.
4. Удобный интерфейс.

5. Подходит для программирования на языке Java.

Основным недостатком является наличие платной версии. Однако также существует бесплатная версия. Кроме этого, для студентов доступна возможность получения платной лицензии бесплатно.

Android Studio – это среда разработки от создателей операционной системы, компанией Google, базирующаяся на основе IntelliJ IDEA [12]. Изначально, она мало чем отличалась от последней.

Кроме того, что данная среда является основной для разработки приложений по ОС Android, ей также присущи следующие преимущества:

1. Бесплатная.
2. Обладает всеми преимуществами IntelliJ IDEA.
3. Адаптирована исключительно под ОС Android, предоставляя удобный набор инструментов для данной системы, например, таких, которые позволяют проводить логирование и профилирование.

Вдобавок к колоссальным преимуществам Android Studio на конечный выбор также повлияли:

1. Эксплуатация данной среды при изучении разработки под ОС Android.
2. Использование IntelliJ IDEA, имеющей идентичный пользовательский интерфейс и сочетание клавиш, при разработке под Java.
3. Применение Android Studio другими разработчиками внутри команды.

3.1.2. Обоснование выбора языка программирования

При выборе языка разработки под определенную мобильную платформу в целом программисту необходимо определить, будет ли приложение являться нативным или нет. Исходя из этого выделяют конкретные языки для разработки. В «Camfrog» существуют отдельные нативные приложения под каждую из мобильных платформ.

Для разработки нативного приложения под ОС Android существует два языка: Java и Kotlin. В таблице 1 представлено обоснование выбора языка программирования.

Таблица 1 – Сравнение языков программирования для ОС Android

Параметры	Вес	Java	Kotlin
Простота разработки	0,3	0,25	0,3
Простота изучения	0,3	0,25	0,3
Распространенность языка	0,2	0,2	0,05
Скорость компиляции	0,1	0,1	0,05
Независимость компонентов разработки от версии ОС Android	0,1	0	0,1
Итого:	1	0,8	0,8

Анализ продемонстрировал, что итоговые показатели данных языков равны. С Kotlin проще процессы разработки и изучения. Java является старшим языком, из-за этого более распространенным и, как следствие, имеет большее сообщество, что способствует более простому поиску нужной информации. Кроме всего прочего, оба языка являются официально поддерживаемым ОС Android.

Более значимым фактором является то, что необходимо провести именно оптимизацию существующего решения, а не разработку с нуля. В текущей реализации используется язык программирования Java. И хотя Kotlin удобнее в разработке, переход на него является длительным и дорогостоящим. Проведение оптимизации с применением Kotlin возможно, но недопустимо, так как в рамках приложения если и подразумевается использование данного языка, то послойно, а не применительно к конкретным модулям или решениям.

3.2. Описание реализации решения

Для осуществления подхода с Single Activity необходимо наследовать все существующие окна не от операции, а от базового компонента-фрагмента. Кроме того, последний должен максимально приближенно реализовать логику из первого. Применительно к приложению «Camfrog», используемому для отображения экрана оба этих компонента, необходимо предварительно перенести всю бизнес-логику из операции (если она по какой-то причине там присутствует) во фрагмент. Вследствие того, что последний не может существовать вне первого, необходимо добавить возможность его открытия из корневого компонента (операции-точки входа).

Сложность зависит от размеров проекта, что обусловлено несколькими причинами.

Во-первых, для небольших приложений, содержащих всего несколько простых окон, достаточно просто выявить требования к новому, заменяющему существующий базовому компоненту отображения пользовательского интерфейса. Что касается крупных, – сделать это не так просто в силу наличия большого числа экранов, которые часто имеют свои особенности (такие, например, как полноэкранность). Из-за этого требования к компоненту постепенно добавляются по мере итеративного перевода логики окон на новый подход, а сам он модифицируется и разрастается.

Во-вторых, процесс перехода требует большого объема ресурсов, в том числе временных. В силу большого числа окон реализация подхода с Single Activity невозможно в быстрые сроки, которые складываются не только из времени на разработку, но и на тестирование и исправление ошибок. По этой причине независимо и разом провести данное изменение попросту невозможно. Это связано с тем, что помимо этой задачи существует и ряд других, которые требуют меньшего времени для реализации и, следовательно, быстрее попадающие в актуальную версию приложения. Попытка объединения всех накопившихся подобных изменений и преобразований, связанных с оптимизацией, приведут к конфликту версий кода, которые просто невозможно

разрешить. Из-за этого данная глобальная задача оптимизации системы навигации декомпозируется в более мелкие, подразумевающие переходы итеративно, от одного окна к другому.

По результате разработки были удовлетворены все прецеденты, описанные ранее при UML-моделировании. Программист может с легкостью манипулировать стеком экранов, по совместному требованию заказчика и дизайнера изменять анимацию переходов между экранами, а также с легкостью контролировать все то, что ранее управлялось флагами операции, с помощью переопределения в коде флагов базового компонента.

Реализация флагов операции изначально внешне выглядела примерно одинаково для всех случаев. На деле же все получилось иначе. Одни требовали добавления в реализацию базового компонента-фрагмента дополнительного поля. Их примерами являются флаги, отвечающие за прозрачность элемента Status Bar и блокировку экрана. Для контроля реакции на поведение клавиатуры стал использоваться отдельный компонент. Способ отображения окна, а конкретнее возможность показа в качестве диалога, и вовсе потребовала почти идентичной реализации еще одного базового компонента. Отличие заключается в разных родительских компонентах Android ОС. И хотя один из данных компонентов наследует другой, применение более конкретного во всех окнах подряд недопустимо, так как, в большинстве случаев, будет присутствовать избыточность.

На рисунках 19, 20 представлена интерпретация различного способа отображения окна.

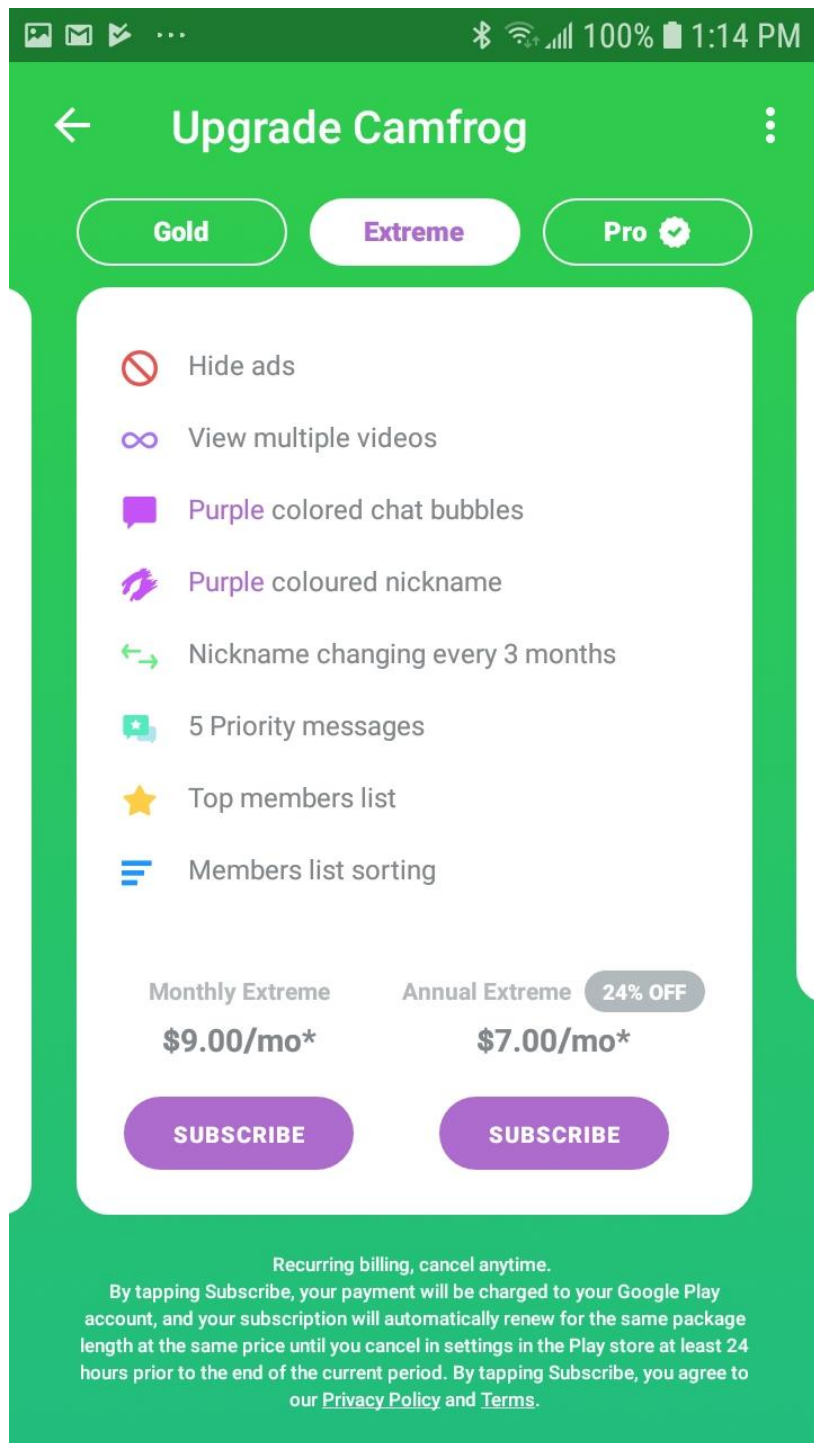


Рисунок 19 – Окно со списком доступных подписок на телефоне

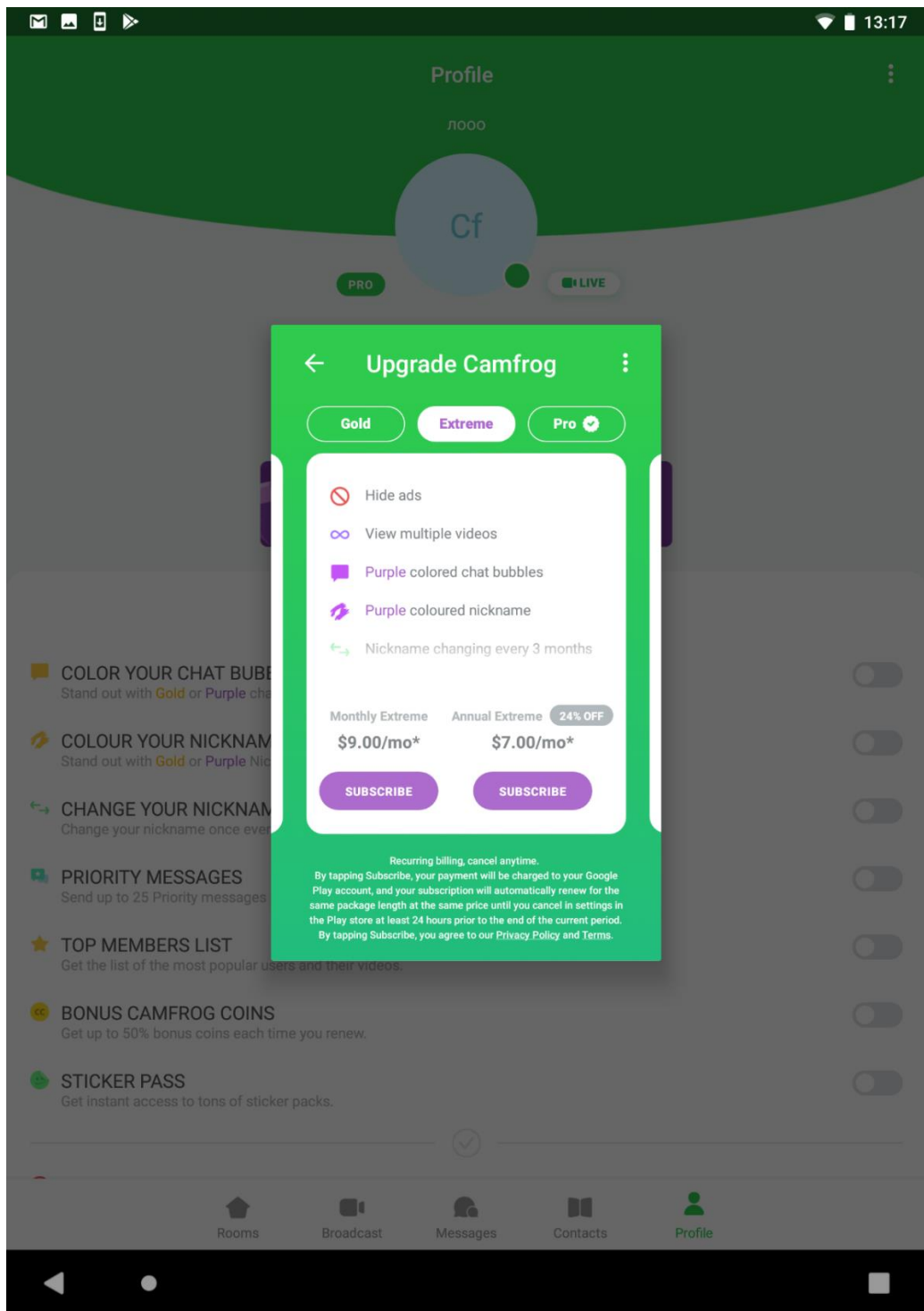


Рисунок 20 – Окно со списком доступных подписок на планшете

Для графического изображения навигации можно применить подход с использованием раскадровки. Раскадровка (Storyboard) – визуальное представление пользовательского интерфейса, отображающее окна с содержимым, а также связи между этими окнами.

В настоящее время Android Studio предоставляет возможность работы с ней, но данные средства разработки на текущий момент не используются в силу их нахождения в тестовом варианте. По этой причине раскадровка была изображена вручную.

Данная процедура необходима для демонстрации изменений в рамках оптимизации системы навигации. Далее представлен ряд изображений раскадровки (рис. 21 – рис. 25), характерный для приложения после проведения оптимизации.

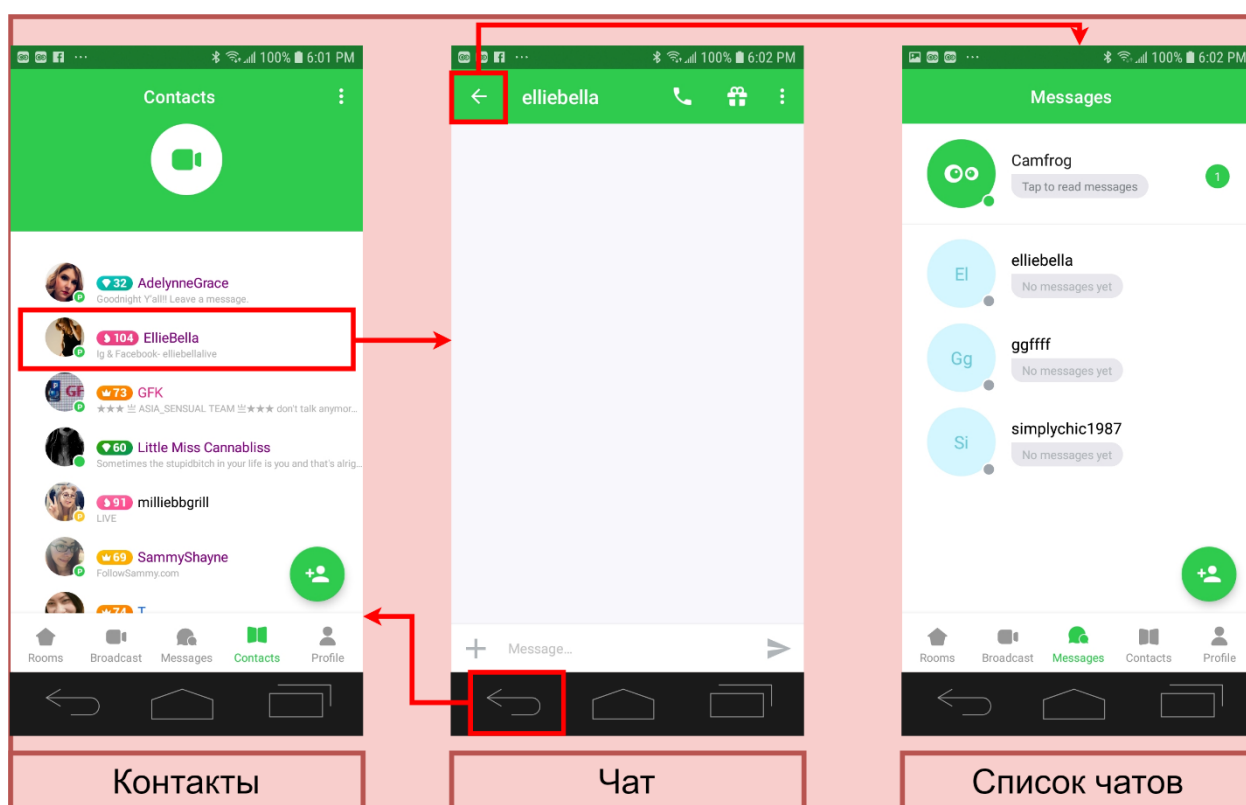


Рисунок 21 – Раскадровка «Контакты – Чат – Список чатов»

При нажатии на чей-либо контакт из списка контактов будет открыт чат с данным пользователем. Нажатие на системную кнопку «Back» ведет на предыдущее окно, а на «Up» – на список всех чатов. Ранее последнее поведение было недостижимо и перед списком всех чатов для пользователя отображалось окно с контактами.

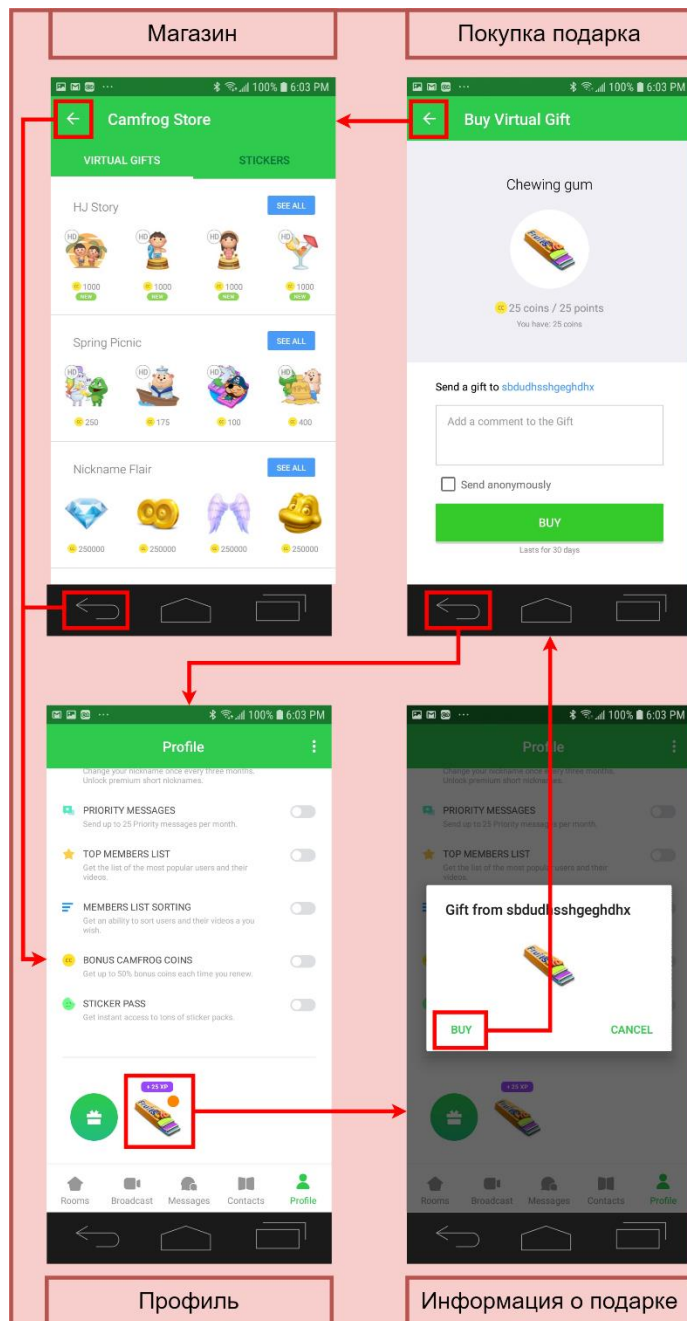


Рисунок 22 – Раскадровка «Профиль – Покупка подарка»

Похожий на описанный ранее случай также имеет место быть между окнами профиля и покупки подарка. До проведения оптимизации при нажатии на кнопку «Buy» с диалогового окна информации о подарке сначала открывался магазина, а после – покупки конкретного подарка. Кроме этого, нажатие системной кнопки «Back» из последнего экрана вело в магазин, а не в профиль.

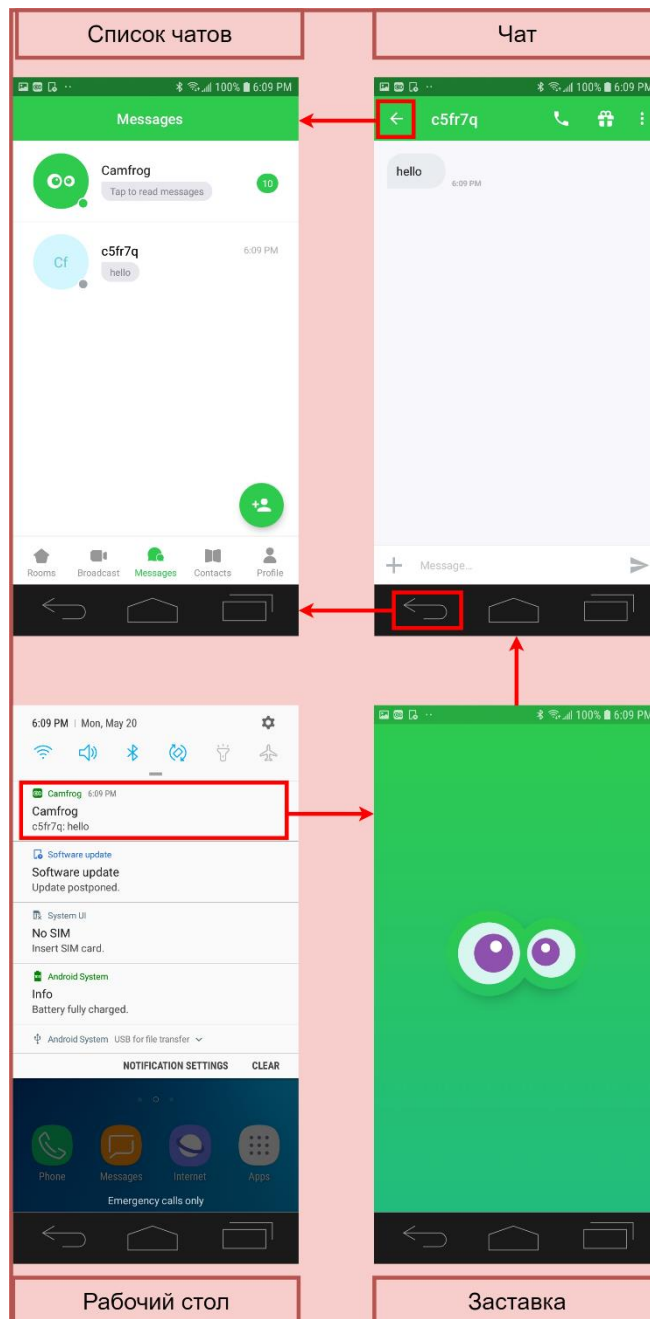


Рисунок 23 – Раскадровка «Рабочий стол – Чат – Список чатов»

Ранее, в обзоре существующего решения, была подробно описана проблема с множественной точкой входа. До оптимизации при разворачивании сообщения от пользователя чат открывался через список всех чатов. Последний является частью главного окна, под которым ничего в стеке не должно находиться. Если пользователь разворачивал уведомление находясь в каком-то окне приложения, обратно вернуться он не мог.

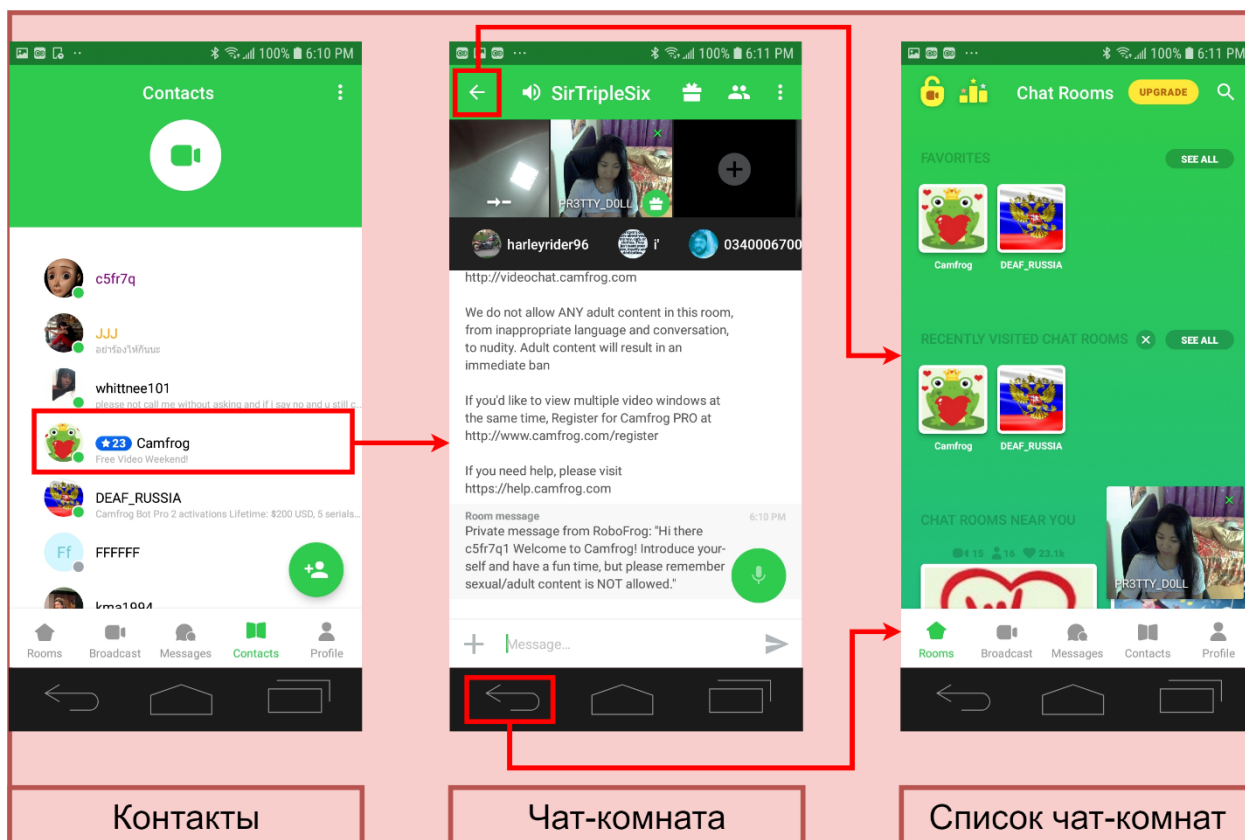


Рисунок 24 – Раскадровка «Контакты – Чат-комната – Список чат-комнат»

В рамках бизнес-логики приложения «Camfrog» в списке контактов могут находиться как пользователи, так и чат-комнаты. Если нажатие на пользователя ведет в чат с ним, то нажатие на чат-комнату открывает ее. В соответствии с нефункциональными требованиями к мобильному приложению применительно к медиа источникам, они должны располагаться в стеке непосредственно за списком данного типа медиа. До проведения оптимизации при выходе с чат-комнаты с помощью кнопок «Up» или «Back» сначала отображался источник ее открытия, а после этого окно менялось на список чат-комнат.

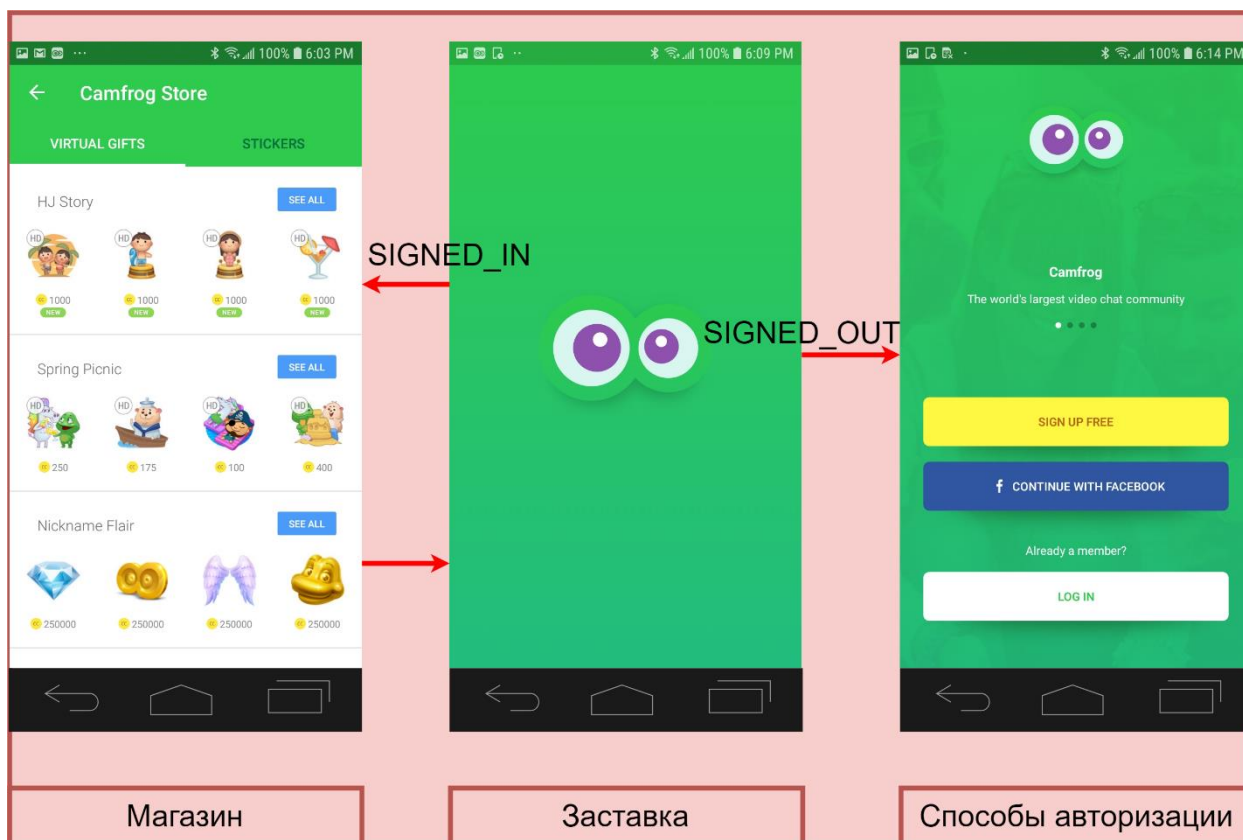


Рисунок 25 – Раскадровка открытия приложения с фонового режима

Одним из сложных в реализации моментов является открытие приложения из фонового режима, когда оно уже было запущено, но длительное время никак не задействовалось. В таком случае требуется перезапрос подключения к серверу. Для успешного подключения характерно закрытие заставки и отображение предыдущего окна; в ином случае – полная чистка стека и отображение способов авторизации. Сложность заключается в анимации перехода из заставки в способы авторизации, подразумевающей последовательное выдвигание элементов. Технически, элементы стека могут удаляться исключительно с вершины. Поэтому удаление заставки из стека неизбежно. Но, в таком случае, анимация перехода не произойдет. Для этого в случае отрицательной попытки подключения сначала полностью чистится стек, а после незаметно для пользователя открывается идентичная заставка, в итоге запускающая анимацию перехода.

3.3. Использование измененной системы

В качестве примера использования оптимизированной системы навигации на рисунке 26 представлена диаграмма классов.

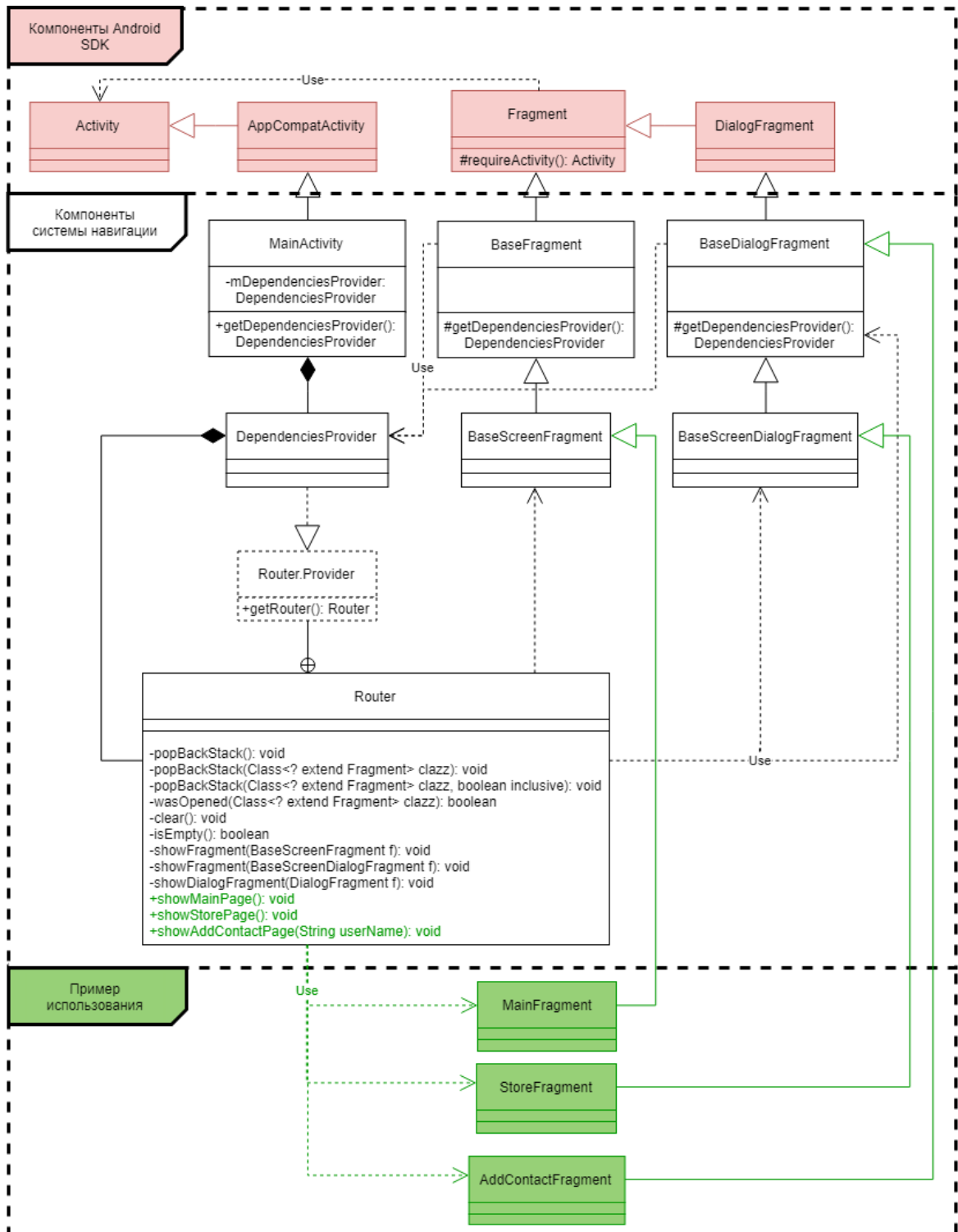


Рисунок 26 – Диаграмма классов системы навигации

Более детально данную диаграмму раскрывает пояснение к ней, представленное в таблице 2.

Таблица 2 – Пояснение к диаграмме классов

Класс	Родитель	Описание
Компоненты Android SDK		
Activity		Компонент приложения, который выдает экран для взаимодействия с пользователем.
AppCompatActivity	Activity	Базовый компонент Activity с поддержкой элемента ActionBar стандартной support library.
Fragment		Динамическая часть пользовательского интерфейса в Activity.
DialogFragment	Fragment	Обертка над Fragment для отображения диалоговых окон.
Компоненты системы навигации		
MainActivity	AppCompatActivity	Точка входа в приложение. В данный класс приходят действия по дальнейшей навигации из внешних источников.
BaseFragment	Fragment	Наследник Fragment, вызывающий дополнительные методы, оборачивая существующие методы жизненного цикла. Содержит базовый функционал динамической части пользовательского интерфейса.
BaseDialogFragment	DialogFragment	Наследник DialogFragment, вызывающий дополнительные методы, оборачивая существующие методы жизненного цикла. Содержит базовый функционал диалоговой динамической части пользовательского интерфейса.
BaseScreenFragment	BaseFragment	Обертка над BaseFragment, инкапсулирующая работу с окном. Наследники – окна, участвующие в навигации, которые в обязательном порядке отображаются в полноэкранном режиме.

Продолжение таблицы 2

Класс	Родитель	Описание
Компоненты системы навигации		
BaseScreenDialogFragment	BaseDialogFragment	Обертка над BaseDialogFragment, инкапсулирующая работу с окном. Наследники – окна, участвующие в навигации, которые могут отображаться как в полноэкранном режиме, так и в диалоговом.
DependenciesProvider	Router.Provider	Класс, хранящий и предоставляющий необходимые сущности (в данном случае, Router).
Router		Класс, инкапсулирующий логику навигации между окон.
Примеры использования		
MainFragment	BaseScreenFragment	Главное окно приложения.
StoreFragment	BaseScreenDialogFragment	Окно магазина, которое показывается как диалог, если используемый девайс – планшет, и в полноэкранном режиме, если используется телефон.
AddContactFragment	BaseDialogFragment	Диалоговое окно с формой для ввода имени пользователя для добавления в список контактов.

3.4. Вывод по главе

Данная глава демонстрирует обоснование средств разработки, детали реализации, внешние отличия после всех внесенных изменений, а также диаграмму классов с пояснительной таблицы к ней, раскрывающих основные компоненты .

Глава 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Целью данной работы является проведение оптимизации системы навигации мобильного приложения для увеличения ее гибкости.

Данный раздел предназначен для экономического обоснования достижения поставленной цели через разработку собственного решения, а также выявления финансовых вложений в данный процесс.

4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

4.1.1. Потенциальные потребители результатов исследования

Для обоснования экономической значимости решения необходимо иметь представление о его сути, актуальности и потребителях.

Проблемой существующего решения является недостаточная его гибкость для решения задач заказчика.

Актуальность заключается в сути оптимизации. Она направлена не только на решение проблемы, но и на минимизацию затрат заказчика при добавлении новой функциональности, связанной с навигацией. Это обусловлено тем, что текущая реализация не является однозначной и удобной в использовании, из-за чего разработчик затрачивает больше времени, чем потенциально мог бы.

Оптимизированный модуль навигации в первую очередь предполагает под собой использование командами мобильной разработки под ОС Android приложений «Camfrog» и «Paltalk». В дальнейшем потенциально возможно его применение как другими разрабатываемыми компанией проектами, так и сторонними приложениями вследствие выделения модуля в отдельную библиотеку.

4.1.2. Анализ конкурентных технических решений

Выявленная проблема не является уникальной, неповторимой, с которой ранее никто не встречался. Некоторые сторонние команды мобильной разработки предоставляют по их меркам гибкие решения в области навигации. Вследствие этого необходимо определить конкурентоспособные черты разработки собственного решения.

Наиболее популярными готовыми решениями являются такие библиотеки, как «Cicerone» и «Conductor». На их официальных страницах представлены данные по масштабам применимости, а также некоторые использующие данные решения приложения.

Для оценки конкурентоспособности необходимо учесть следующие факторы:

1. Простота использования решения – для минимизации необходимых человеко-часов при разработке;
2. Применение стандартного подхода – для легкости освоения и понимания того, что происходит внутри модуля;
3. Легкость расширения – для возможного добавления нестандартных надстроек.
4. Однозначность работы со стеком – для простоты понимания и возврата к уже открытым окнам.
5. Необходимость начальной разработки – для определения потребности в первичных финансовых вложениях перед применением решения.

В таблице 3 представлена оценочная карта сравнения конкурентов, основанная на вышеизложенных данных.

Таблица 3 – Оценочная карта сравнения конкурентных технических решений

Конкуренты		Cicerone	Conductor	Свое решение	b_j	w_j
Факторы конкурентоспособности	Простота использования	8 1,6	5 1	10 2	4	0,2
	Применение стандартного подхода	7 1,75	3 0,75	10 2,5	4	0,25
	Легкость расширения	8 2,4	5 1,5	9 2,7	5	0,3
	Однозначность работы со стеком	8 1,2	4 0,6	8 1,2	3	0,15
	Необходимость начальной разработки	10 1	10 1	0 0	2	0,1
Итоговая оценка		7,95	4,85	8,4	20	1

Для большей наглядности данные таблицы отображены в виде многоугольника конкурентоспособности на рисунке 27, на котором представлены указанные факторы применительно к каждому из решений.

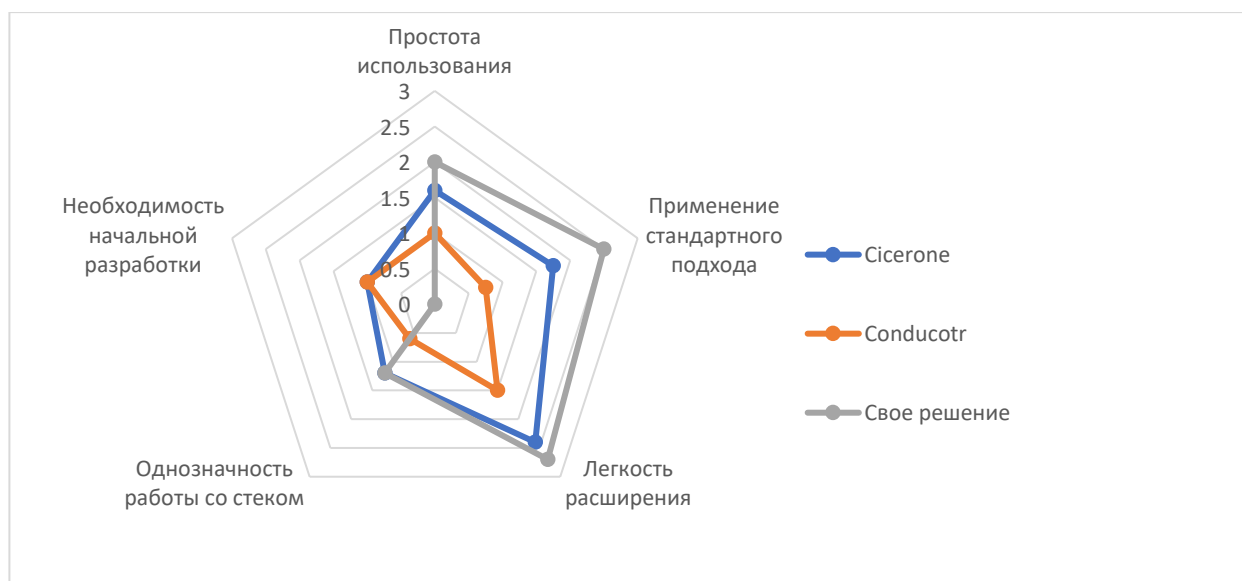


Рисунок 27 – Многоугольник конкурентоспособности

Данные по конкурентоспособности демонстрируют доминирование использования собственного решения над конкурентами (за исключением необходимости начальной разработки). Это связано с тем, что при создании своей реализации программист (и, как следствие, команда разработки в целом)

закладывает в основу то, что считает наиболее значимым. Проблема существующего решения заключается в недостаточной гибкости, следовательно наиболее значимым фактором при проведении оптимизации является «Легкость расширения».

4.1.3. SWOT-анализ

Для оценки внутренних и внешних факторов разрабатываемого решения существует метод оценки текущей ситуации и будущих перспектив – SWOT-анализ. Данный подход направлен на выявление ряда действий, необходимого для качественного, эффективного достижения поставленной цели. Кроме того, он помогает в структурированном описании проведения оптимизации модуля навигации мобильного приложения в целом [13].

В результате выполнения анализа выявляются всевозможные сочетания различных факторов внешней среды и внутренних свойств решения. Исходя из этого выделяются особенности, которые должны быть учтены на этапе разработки для максимизации положительных сторон и минимизации отрицательных. Анализ также показывает, насколько достижимы реализация имеющихся возможностей и противостояние внешним угрозам.

В ходе данного этапа работы был проведен SWOT-анализ оптимизации модуля навигации мобильного приложения через разработку собственного решения, результаты которого представлены в таблице 4.

Таблица 4 – Сводная матрица SWOT-анализа

	<p>Сильные стороны: С1. Расширяемость С2. Подконтрольность С3. Сопровождение документацией С4. Легкость освоения и применения С5. Строгое следование подходу с Single Activity</p>	<p>Слабые стороны: Сл1. Длительные временные затраты на разработку Сл2. Дополнительные финансовые затраты в разработку Сл3. Выявление новых требований по ходу разработки</p>
<p>Возможности: В1. Применение решения внутри всех проектов компании для мобильных приложений под ОС Android В2. Выделение решения в отдельную библиотеку В3. Анализ переходов пользователей внутри приложения</p>	<p>Расширяемость решения позволяет добавить новую функциональность без больших трудозатрат. Сопровождение документацией, легкость освоения и применения решения делает возможным его использование в любых мобильных приложениях ОС Android.</p>	<p>Дальнейшее переиспользование решения в других проектах компании способно «перекрыть» финансовые затраты, так как в этом случае происходит экономия за счет разработки и сопровождения отдельных реализаций. Нарастивание требований использующими библиотеку приложениями ведет к созданию более общей структуры, подходящей под любые требования.</p>
<p>Угрозы: У1. Ошибки в поведении У2. Появление более современного стандартного подхода к разработке навигации в ОС Android У3. Отказ заказчика от создания собственного решения в рамках пересмотра планирования</p>	<p>Так как собственное решение находится под полным контролем команды разработки (даже базовые компоненты доступны к редактированию), исправление потенциально возможных ошибок в поведении вследствие любых причин является возможным и, как правило, не занимает большой объема времени.</p>	<p>Основанные на выявленных в ходе разработки требований изменения в редких случаях могут кардинально изменять базовые компоненты, вследствие чего образуются новые ошибки. Более быстрый процесс их исправления требует внутреннего версионирования изменений на уровне отделов тестирования и разработки.</p>

Результат SWOT-анализа показал, что:

1. Все изложенные возможности могут быть достигнуты за счет сильных сторон;

2. Использование возможностей тем или иным образом сглаживает все имеющиеся внутренние слабости;

3. Избавление от угроз не является возможным. Однако ошибки поведения могут быть максимально сглажены за счет применения сильных сторон и разрешения одной из слабых.

4.2. Планирование научно-исследовательских работ

Для конкретизации структуры работ по достижению поставленной цели необходимо проведение процесса планирования. Правильно структурированная работа и грамотно использованное время сопутствуют положительному результату [14].

4.2.1. Структура работ в рамках научного исследования

На начальном этапе планирования нужно выявить список задействуемых лиц, а также перечень всех работ, подлежащих выполнению в рамках проведения оптимизации. Кроме того, необходимо сопоставить исполнителей с конкретными работами.

Перечень основных этапов, содержание работ и распределение исполнителей представлен в таблице 5.

Таблица 5 – Перечень работ и распределение исполнителей

Основные этапы	№	Содержание работ	Должность исполнителя
Трудовая деятельность по выполнению задания в рамках компании	1	Постановка задания	Руководитель команды мобильной разработки
	2	Подбор и изучение литературы по заданию	Разработчик
	3	Анализ предметной области	Разработчик
	4	Перепроектирование существующего решения	Разработчик

Продолжение таблицы 5

Основные этапы	№	Содержание работ	Должность исполнителя
Трудовая деятельность по выполнению задания в рамках компании	5	Проведение оптимизации посредством разработки собственного решения	Разработчик
	6	Тестирование поведения приложения в области навигации	Тестировщик компании
	7	Доработка решения	Разработчик
	8	Отчет о проделанной работе	Руководитель команды мобильной разработки, разработчик
Выполнение бакалаврской работы	9	Выбор научного руководителя	Разработчик
	10	Утверждение темы бакалаврской работы	Научный руководитель, разработчик
	11	Составление календарного плана выполнения бакалаврской работы	Научный руководитель, разработчик
	12	Согласование основной части проделанной работы с научным руководителем	Научный руководитель, разработчик
	13	Выполнение частей экономического менеджмента и социальной ответственности	Разработчик
	14	Подведение итогов, оформление бакалаврской работы	Разработчик

Данная работа выполняется в рамках трудовой деятельности в должности инженера-разработчика компании ООО «ТомскСофт». По этой причине всю структуру по достижению цели можно разбить на два больших этапа:

1. Реализация программного решения, которая проходит в рамках рабочего процесса и кроме студента-разработчика также частично задействует иных исполнителей компании.

2. Выполнение бакалаврской работы, происходящая в рамках учебной деятельности.

4.2.2. Определение трудоемкости выполнения работ, разработка графика проведения научного исследования

Одной из основных составляющих стоимости разработки являются затраты на трудовую деятельность. Исходя из этого необходимо определить трудоемкость работ каждого действующего лица, участвующего в решении проблемы гибкости модуля навигации мобильного приложения.

Данный показатель носит вероятностный характер, так как зависит от множества факторов. Его оценка производится экспертным путем в человеко-днях. Для определения среднего значения трудоемкости $t_{ожі}$ используется следующая формула:

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5},$$

где $t_{ожі}$ – ожидаемая трудоемкость выполнения работы;

$t_{\min i}$ – минимальная трудоемкость выполнения работы;

$t_{\max i}$ – максимальная трудоемкость выполнения работы.

Основываясь на средней трудоемкости работ, необходимо определить продолжительность каждой работы в рабочих днях T_p с учетом параллельности ее выполнения сразу несколькими исполнителями. Данный показатель рассчитывается по следующей формуле:

$$T_{p_i} = \frac{t_{ожі}}{Ч_i},$$

где T_{p_i} – продолжительность работы в рабочих днях;

$t_{ожі}$ – ожидаемая трудоемкость выполнения работы;

$Ч_i$ – численность параллельно выполняющих работу исполнителей.

При построении графика проведения научной работы необходимо учесть выходные и праздничные дни. Для этого продолжительность каждой работы преобразуется в похожий показатель с учетом календарных дней по следующей формуле:

$$T_{ki} = T_{pi} \cdot k_{\text{кал}},$$

где T_{ki} – продолжительность работы в календарных днях;

T_{pi} – продолжительность работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Для определения последнего коэффициента необходимо использовать следующую формулу:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}},$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году [15].

Для различных основных этапов работ коэффициент календарности будет варьироваться:

1. Для этапа трудовой деятельности на предприятии для разработчика (студента), как и для всех остальных действующих лиц, характерна пятидневная рабочая неделя. Согласно производственному календарю, 2019 год представлен 365 календарными днями, в числе которых 247 рабочих и 118 праздничных и выходных [16]. Исходя из этого коэффициент календарности составляет 1,48.

2. В рамках выполнения бакалаврской работы для студента, как и для научного руководителя, характерна шестидневная рабочая неделя. Для такого рабочего графика производственный календарь представлен 299 рабочими днями и 66 выходными и праздничными [16]. Основываясь на данных сведениях коэффициент календарности равен 1,22.

Описанные показатели по каждой работе представлены в сводной таблице 6.

Таблица 6 – Временные показатели проведения научного исследования

Наименование работы	Исполнители работы	Трудоемкость работ, чел-дни			Длительность работ, дни	
		t _{min}	t _{max}	t _{ож}	T _p	T _к
Постановка задания	Руководитель команды мобильной разработки	1	2	1,4	1	1
Подбор и изучение литературы по заданию	Разработчик	5	10	7	7	10
Анализ предметной области	Разработчик	4	7	5,2	5	7
Перепроектирование существующего решения	Разработчик	15	23	18,2	18	27
Проведение оптимизации посредством разработки собственного решения	Разработчик	97	146	116,6	117	173
Тестирование поведения приложения в области навигации	Тестирующий компании	15	23	18,2	18	27
Отчет о проделанной работе	Руководитель команды мобильной разработки	1	2	1,4	1	1
	Разработчик	1	2	1,4	1	1
Выбор научного руководителя	Разработчик	1	2	1,4	1	1
Утверждение темы бакалаврской работы	Научный руководитель	2	4	2,8	1	1
	Разработчик	2	4	2,8	1	1
Составление календарного плана выполнения бакалаврской работы	Научный руководитель	1	2	1,4	1	1
	Разработчик	3	5	3,8	2	2
Согласование основной части проделанной работы с научным руководителем	Научный руководитель	1	2	1,4	1	1
	Разработчик	2	5	3,2	2	2
Выполнение частей экономического менеджмента и социальной ответственности	Разработчик	5	10	7	7	9
Подведение итогов, оформление бакалаврской работы	Разработчик	5	8	6,2	6	7

Данная таблица демонстрирует, что наиболее длительным задачей является непосредственная разработка, которая занимает порядка 173 календарных дней. Это связано с объемами оптимизируемого приложения. Стоит также заметить, что минимальная и максимальная трудоемкости работ, связанных с перепроектированием, разработкой, тестированием и доработкой, являются взаимозависимыми, и относятся друг к другу как 1:1,5. Это обусловлено спецификой планирования времени, запрашиваемого у заказчика на выполнение тех или иных задач.

Наглядное представление временных показателей отображено в качестве диаграммы Ганта в таблице 1 приложения А.

Диаграмма показывает возможность выполнения работ параллельно не только разными исполнителями, но также и одним и тем же. Такое поведение является доступным из-за возможности комбинирования задач, одновременного выполнения сразу нескольких работ. Так, например, анализ предметной области в ходе изучения литературы является не просто возможным, но также и более продуктивным, так как данные работы по своей сути закрепляют друг друга. Также это приводит к экономии временных ресурсов.

С учетом параллельности, суммарная длительность выполнения работ в календарных днях каждым членом рабочей группы распределилась следующим образом:

1. Руководитель команды мобильной разработки – 2 (0,7%).
2. Научный руководитель – 3 (1,0%).
3. Тестировщик компании – 27 (9,4%).
4. Разработчик – 255 (88,9%).

4.2.3. Бюджет научно-технического исследования

Бюджетное планирование – это инструмент для реализации стратегических планов. Эффективно работающие системы бюджетирования не только, кардинально облегчают планирование и координацию деятельности, но

и делают прозрачными все мероприятия и расходуемые ими ресурсы, что существенно повышает эффективность работ [17].

4.2.3.1. Расчет материальных затрат научно-технического исследования

При рассмотрении материальных затрат необходимо рассмотреть материалы, используемые для достижения поставленной цели. В рамках работы учтены расходы исключительно на канцелярские принадлежности, предназначенные для удобного и привычного составления графиков и диаграмм, а также на распечатку текстового материала. Непосредственная разработка не несет материальных затрат, так как в ходе нее не используются физические материалы.

Список сопутствующих работе средств и расчет материальных затрат представлен в таблице 7.

Таблица 7 – Материальные затраты

Наименование материала	Единица измерения	Количество	Цена за единицу материала, руб.	Затраты, руб.
Лист с распечатанным текстовым материалом формата А4	шт.	250	1,90	475,00
Шариковая ручка	шт.	1	73,50	73,50
Тетрадь в клетку 96 листов	шт.	1	37,50	37,50
Итого:				586,00

4.2.3.2. Расчет затрат на специальное оборудование

Для проведения оптимизации модуля мобильного приложения отдельного специального оборудования не приобреталось. Используемое оборудование представлено персональным компьютером и мобильным телефоном на базе ОС Android.

Для расчета затрат на имеющееся специальное оборудование необходимо рассчитать его амортизацию. Начальные стоимости персонального компьютера и мобильного телефона составляли 72850 и 13200 рублей соответственно. Сроки полезного использования данного оборудования составляют 2-3 года и 3-5 лет соответственно. Для выполнения поставленной задачи планируется эксплуатация данной техники в течение 9 месяцев.

Произведенный расчет затрат на амортизацию, представлен в таблице 8.

Таблица 8 – Затраты на амортизацию.

Наименование оборудования	Норма амортизации, %	Амортизационные отчисления, руб.	
		ежемесячные	суммарные
Персональный компьютер	33,33%	2003,38	18030,42
Мобильный телефон	20%	220,00	1980,00
Итого:			20010,42

4.2.3.3. Основная заработная плата исполнителей

Основной заработной платой называется вознаграждение за выполненную работу в соответствии с установленными нормами труда. Она рассчитывается на основе трудоемкости и действующей системы окладов исполнителей. Для расчета основной заработной платы используется следующая формула:

$$Z_{осн} = Z_{дн} \cdot Tр \cdot (1 + K_{пр} + K_{д}) \cdot K_{р},$$

где $Z_{дн}$ – среднедневная заработная плата;

$Tр$ – продолжительность работ, выполняемых работником;

$K_{пр}$ – премиальный коэффициент, составляющий 0,3;

$K_{д}$ – коэффициент доплат и надбавок, равный 0,35;

$K_{р}$ – районный коэффициент, который для города Томска составляет 1,3.

Показатель среднедневной заработной платы является составным и рассчитывается по формуле:

$$Z_{дн} = \frac{Z_{м} \cdot M}{F_{д}},$$

где Z_m – месячный оклад исполнителя;

M – количество месяцев работы без отпуска, составляющий 11,2 и 10,4 месяца для пятидневной и шестидневной недель соответственно;

F_d – действительный годовой фонд рабочего времени персонала.

Расчет последнего показателя для всех действующих лиц представлен в таблице 9.

Таблица 9 – Баланс рабочего времени

Показатели рабочего времени	Руководитель команды мобильной разработки	Научный руководитель	Тестировщик	Разработчик
Календарное число дней	365	365	365	365
Количество нерабочих дней	118	66	118	118
Потери рабочего времени	30	56	30	30
Действительный годовой фонд рабочего времени	217	243	217	217

При расчете основной заработной платы следует учесть следующие показатели окладов исполнителей:

1. Руководитель команды мобильной разработки – 64527,00 рублей;
2. Научный руководитель – 33664,00 рубля;
3. Тестировщик – 38925,00 рублей;
4. Разработчик – 25287,00 рублей.

Результаты расчета представлены в таблице 10.

Таблица 10 – Расчет основной заработной платы

Исполнители	Здн, руб.	Кпр	Кд	Кр	Тр	Зосн
Руководитель команды мобильной разработки	3330,43	0,3	0,35	1,3	2	14287,53
Научный руководитель	1440,76	0,3	0,35	1,3	3	9271,31
Тестировщик	2009,03	0,3	0,35	1,3	18	77568,74
Разработчик	1305,14	0,3	0,35	1,3	175	489915,23
Итого:						591042,81

4.2.3.4. Дополнительная заработная плата исполнителей

Дополнительной заработной платой считаются денежные выплаты компенсационного, стимулирующего или разового характера. Данный показатель вычисляется произведением основной заработной платы и коэффициента дополнительной. Как правило на этапе проектирования данный показатель составляет от 0,12 до 0,15. Для данной работы был его значение было принято в 0,13.

Исходя из этого были выявлены дополнительный заработные платы исполнителей, представленные в таблице 11.

Таблица 11 – Результаты расчета дополнительной заработной платы

Руководитель команды мобильной разработки	Научный руководитель	Тестировщик	Разработчик	Итого
1857,38	1205,27	10083,94	63688,98	76835,57

4.2.3.5. Отчисления во внебюджетные фонды

Отчисления во внебюджетные фонды в Российской Федерации учитываются выплаты в фонд социального страхования, пенсионный фонд, а также в федеральный фонд обязательного медицинского страхования.

Величина отчислений зависит от затрат на оплату труда работников и выражается следующей формулой:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}}),$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды;

Для учреждений, осуществляющих образовательную и научную деятельность, а также организаций, реализующих деятельность в области информационных технологий в 2019 году, вводится пониженная ставка, равная 28% [18].

Размер этих отчислений для исполнителей представлен в таблице 12.

Таблица 12 – Отчисления во внебюджетные фонды

Руководитель команды мобильной разработки	Научный руководитель	Тестировщик	Разработчик	Итого
4520,57	2933,44	24542,75	155009,18	187005,95

4.2.3.6. Накладные расходы

Для учета дополнительных затрат, не относящихся к основному производству, используются накладные расходы. Данный показатель рассчитывается произведением суммы учтенных ранее расходов на коэффициент накладных расходов, который составляет порядка 16%.

Для данной работы накладные расходы составляют 140076,92 рублей.

4.2.3.7. Формирование бюджета затрат научно-исследовательского проекта

Бюджет разрабатываемого решения базируется на основе рассчитанной величины затрат. Его значение представляет нижнюю границу реальных затрат, необходимых для достижения поставленной цели.

Расчет бюджета затрат НИП представлен в таблице 13.

Таблица 13 – Расчет бюджета затрат НИП

Наименование	Сумма, руб.	Удельный вес, %
Материальные затраты	586,00	0,06
Затраты на специальное оборудование	20010,42	1,97
Затраты на основную заработную плату	591042,81	58,20
Затраты на дополнительную заработную плату	76835,57	7,57
Страховые взносы	187005,95	18,41
Накладные расходы	140076,92	13,79
Общий бюджет	1015557,67	100,00

4.3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

При определении эффективности работы необходимо рассчитать интегральный показатель эффективности научного исследования, нахождение которого связано с определением финансовой эффективности и ресурсоэффективности. Для этого нужно учесть три варианта проведения работ по оптимизации модуля навигации мобильного приложения.

Интегральный финансовый показатель определяется по формуле:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}},$$

где Φ_{pi} – стоимость варианта исполнения;

Φ_{max} – максимальная стоимость исполнения.

Расчет интегрального показателя ресурсоэффективности каждого из вариантов исполнения можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i,$$

где a_i – весовой коэффициент варианта исполнения;

b_i , – балльная оценка варианта исполнения;

n – число параметров сравнения.

Результаты расчета показателя базируются на данных таблицы 14.

Таблица 14. Сравнительная оценка характеристик вариантов исполнения

Критерии	Весовой коэффициент параметра	И1	И2	И3
Возможность переиспользования	0,4	5	3	4
Простота применения	0,2	2	4	1
Расширяемость	0,3	4	3	5
Простота разработки	0,1	2	5	1
I_p, Итого	1	3,8	3,4	3,4

Для нахождения интегрального показателя эффективности вариантов исполнения необходимо найти частное ресурсоэффективности и интегрального финансового показателя.

Сравнение интегрального показателя эффективности вариантов исполнения позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Проведенный расчет данного показателя, а также сравнение вариантов исполнения по нему представлены в таблице 15.

Таблица 15. Сравнительная эффективность разработки

Показатели	И1	И2	И3
Интегральный финансовый показатель разработки	1	1	1
Интегральный показатель ресурсоэффективности разработки	3,8	3,4	3,4
Интегральный показатель эффективности	3,8	3,4	3,4
Сравнительная эффективность вариантов исполнения	1,12	1	1

Данная таблица демонстрирует, что наиболее эффективным вариантом для разработки собственного решения в рамках оптимизации является исполнение 1.

4.4. Вывод по главе

Основываясь на результатах проведенного анализа, разработка собственного решения для оптимизации модуля навигации мобильного приложения ОС Android является конкурентоспособной и перспективной. Предполагается унификация и использование решения и в других проектах заказчика. Длительность разработки составила 263 календарных дня, а рассчитанная стоимость – чуть более 1015 тыс. рублей. Кроме того, был выбран наиболее эффективный вариант разработки.

Глава 5. Социальная ответственность

Оптимизация системы навигации мобильного приложения осуществляется в качестве инженера-разработчика организации ООО «ТомскСофт». Пользователями решения являются внутренние команды разработки приложений под ОС Android.

В рамках производственной деятельности программист сталкивается с различными видами социальной ответственности. Их знание и соблюдение способствует не только комфортной, безопасной работе, но и снижению вероятности возникновения внештатных ситуаций, влекущих негативные последствия.

Данный раздел демонстрирует анализ ряда аспектов, влияющих на безопасность трудовой деятельности.

5.1. Правовые и организационные вопросы обеспечения безопасности

Основным инструментом создания любого программного кода является персональный компьютер. Безопасность его эксплуатации в организационной области регламентируется такими нормативными актами, как ГОСТ 12.2.032-78 ССБТ. «Рабочее место при выполнении работ сидя» и СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Исходя из содержания данных документов, организация такого вида работы, при котором используется персональный компьютер, подразумевает выполнение следующих требований [19,20]:

1. Используемые для осуществления операций средства должны находиться в пределах досягаемости моторного поля. Исходя из этого в рабочем пространстве устройства ввода данных персонального компьютера располагаются в областях часто и очень часто используемых органов управления, а канцелярские принадлежности – в зоне редкого использования.

2. С учетом роста разработчика в 1800 мм конструкция рабочего стола должна отвечать следующим минимальным характеристикам: высота рабочей поверхности – 725мм, пространство для ног – 675мм, глубина на уровне колен и вытянутых ног – 450мм и 650мм соответственно. Реальная конструкция рабочего стола отвечает данным характеристикам.

3. Оптимальная рабочая поза обеспечивается при помощи полумягкого поворотного сиденья с наличием подлокотников, имеющего возможность регулировки угла наклона и высоты. Последняя, вместе с подставкой для ног, не требуется для заданной высоты работника.

4. Расположение экрана предусматривает расстояние между программистом в 600-700мм и линию взгляда ниже горизонтальной на 20 градусов. Это достигается с помощью подвижности видеомонитора, а также возможности регулировки его высоты и угла обзора.

5. Через оконные проемы, оборудованные жалюзи, доступно проникновение естественного света.

6. Площадь рабочего пространства, приходящаяся на одного человека, должна быть более 6м². Длина и ширина помещения составляют 7м и 6м соответственно. С учетом наличия 6 рабочих мест, площадь, приходящаяся на одного разработчика, составляет 7м².

7. Помещения подлежат ежедневной влажной уборке персоналом бизнес-центра; также осуществляется его проветривание в течение рабочего дня.

8. Световые проемы расположены по левую сторону от рабочего места.

9. Для осуществления искусственного освещения используются многоламповые светильники с электромагнитными пускорегулирующими аппаратами.

Обеспечение безопасности рабочего процесса с правовой точки зрения регламентируется Трудовым кодексом Российской Федерации от 30.12.2001 N 197-ФЗ. Данный правовой акт гласит, что [21]:

1. Нормальная продолжительность рабочего времени не может превышать 40 часов в неделю. Исходя из очной учебной деятельности, трудовой

договор составлен с учетом неполного рабочего дня: без ограничения минимальной продолжительности, с наличием верхней границы, предусмотренной Трудовым Кодексом.

2. Перерыв на отдых и питание в течение рабочего дня должен составлять от 30 до 120 минут. В компании предусмотрен один перерыв или более с суммарной продолжительностью в 60 минут.

3. Выплата заработной платы должна производиться не реже, чем раз в полмесяца. С учетом трудового договора, заработная плата за месяц дробится на две составляющие: фиксированную, аванс, предоставляемую в начале месяца, и оставшуюся часть месячной заработной платы, выплачиваемую в конце.

Организация предоставляет сотрудникам корпоративный электронный ресурс с различной информацией в области трудового процесса: от прав и обязанностей в соответствии с занимаемой должностью до составляющих премии.

В ходе решения поставленной проблемы нарушений в области правового и организационного обеспечения безопасности не выявлено.

5.2. Производственная безопасность

В ходе любой производственной деятельности человек сталкивается с различными факторами. По характеру проявления они делятся на вредные, вызывающие заболевания, и опасные, подвергающие травмам.

Мобильная разработка не лишена данных факторов. В ходе работы было проведено наблюдение по их конкретизации в соответствии с ГОСТ 12.0.003-2015 «Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация» [22]. Результат анализа представлен в таблице 16.

Таблица 16 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
1. Электромагнитные излучения	ГОСТ 12.1.006-84 ССБТ. «Электромагнитные поля радиочастот. Общие требования безопасности.» СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно- вычислительным машинам и организации работы»
2. Недостаток естественного освещения	СанПиН 2.2.1/2.1.1.1278–03. «Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий.» СП 52.13330.2016 «Естественное и искусственное освещение.» СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно- вычислительным машинам и организации работы»
3. Эмоциональные перегрузки	Р 2.2.2006-05 «Гигиена труда. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда.»
4. Опасность поражения электрическим током	ГОСТ 12.1.019-2017 ССБТ. «Электробезопасность. Общие требования и номенклатура видов защиты.»

Исходя из данной таблицы было выявлено, что на программиста в ходе его деятельности воздействуют факторы физического и психологического воздействия. Отсутствие химических и биологических факторов связано исключительно со спецификой предметной области. Этапы работ по реализации решения, состоящие из разработки и эксплуатации, подразумевают использование персонального компьютера в обоих случаях.

5.2.1. Электромагнитные излучения

Любая техника, подключенная к бытовой электросети, характеризующейся напряжением близким к 220 В и частотой в 50 Гц, является

источником электромагнитного поля, а то, в свою очередь, – электромагнитных излучений. Конкретикой такой техники, с которой в ходе работы происходило непосредственное взаимодействие, являются системный блок, видеомонитор и мобильный телефон.

Электромагнитные излучения обладают способностью биологического, теплового и нетеплового воздействия на организм человека.

Биологическое воздействие излучений зависит не только от их показателей, таких как длина волны и интенсивность, но и от продолжительности и режимов воздействия, анатомического строения и размеров органов.

Нетепловое воздействие главным образом влияет на центральную и сердечно-сосудистую системы. Происходит нарушение рефлекторной деятельности, снижение активности мозга. Возможны отклонения со стороны эндокринной системы.

Тепловое воздействие характеризуется повышением температуры тела и органов вследствие преобразования излучений в тепловую энергию. Такого рода нагревание может вызвать обострение хронических заболеваний.

В соответствии с нормативными документами, уровень напряженности электромагнитного поля на рабочем месте в течение стандартного восьмичасового рабочего дня не должен превышать 5кВ/м. Измеренные значения данного показателя в среднем составляют 3,8кВ/м, что удовлетворяет требованиям.

Для минимизации воздействия электромагнитных излучений существует ряд решений:

1. Использование внешних защитных фильтров для видеомонитора.
2. Использование качественной электропроводки.
3. Комплексная оценка электромагнитной обстановки с учетом расположения рабочих мест.

5.2.2. Недостаток естественного освещения

Одним из вредных производственных факторов, воздействующих на организм человека, является недостаток естественного освещения. И хотя его переизбыток – потенциальная проблема, ее решением является ограничение через использование жалюзи в рабочем помещении. Добиться же «дополнительного» естественного света достаточно сложно.

Источником фактора недостаточности естественного света является географическое расположение офиса, а также планировка помещений. Несмотря на то, что оконные проемы расположены так, как этого требует СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», по левую сторону от разработчика, окна помещения выходят на юг, что является противоречием рекомендаций.

Солнечные лучи обладают высокой гигиенической и биологической ценностью для организма человека. Они не только несут свет и тепло, но также являются источником ультрафиолетовых излучений. Последние стимулируют обменные процессы, повышают иммунитет. Недостаток ультрафиолетовой радиации снижает работоспособность, устойчивость к заболеваниям. Кроме того, эти лучи способны уничтожать болезнетворные микроорганизмы.

В рабочем помещении используется боковое освещение. С учетом этого в соответствии с СанПиН 2.2.1/2.1.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий» для помещений, оборудованных ЭВМ, коэффициент естественной освещенности в середине данных помещений должен составлять не менее 1,2 %. Задокументированное в общей информации о данном рабочем помещении ранее проведенное измерение (с учетом показателя облачности в 11 баллов) показало, что данный коэффициент составил 36,7 %. Из этого следует, что освещение рабочего места посредством естественного света соответствует нормативному значению.

Снижение влияния естественного света является достаточно сложной задачей. Из-за этого её решение требует кардинальных, весомых действий:

1. Оценка обстановки естественного света с последующей перепланировкой взаимного расположения рабочих мест внутри помещения;
2. Переезд компании в другой, географически более выгодно расположенный офис.

5.2.3. Эмоциональные перегрузки

Работа любого программиста главным образом заключена в умственном труде. Его деятельность направлена на реализацию новых решений, в ходе которой он анализирует большие объемы информации. Таким образом, в течение большей части рабочего времени разработчик задействует свой мыслительный процесс. Это вызывает умственное перенапряжение, что, как следствие, является причиной эмоциональных перегрузок.

Вследствие этого фактора у разработчика снижается работоспособность, повышается раздражительность и конфликтность. Также это способствует развитию профессиональной болезни, при которой пропадает интерес к работе.

Кроме психологического воздействия эмоциональный стресс способствует нарушению деятельности сердца, повышению кровяного давления, развитию заболеваний кожи и кишечного-желудочного тракта.

В таблице 17 приведена классификация условий труда на основе эмоциональной нагрузки в соответствии с Р 2.2.2006-05 «Гигиена труда. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда» [23].

Таблица 17 – Классы условий труда по показателю эмоциональной нагрузки трудового процесса

Показатели напряженности трудового процесса		Степень ответственности за результат собственной деятельности. Значимость ошибки	Степень риска для собственной жизни	Степень ответственности за безопасность других лиц	Количество конфликтных ситуаций, обусловленных профессиональной деятельностью, за смену		
Класс условий труда	оптимальный	Напряженность труда легкой степени	Несет ответственность за выполнение отдельных элементов заданий. Влечет за собой дополнительные усилия в работе со стороны работника	Исключена	Исключена	Отсутствует	
	допустимый	Напряженность труда средней степени	Несет ответственность за функциональное качество вспомогательных работ. Влечет за собой дополнительные усилия со стороны вышестоящего руководства			1-3	
	вредный	Напряженный труд	1 степени	Несет ответственность за функциональное качество основной работы. Влечет за собой исправления за счет дополнительных усилий всего коллектива			4-8
			2 степени	Несет ответственность за функциональное качество конечной продукции, работы, задания. Влечет за собой повреждение оборудования, остановку технологического процесса и может возникнуть опасность для жизни	Вероятна	Возможна	Более 8

С учетом занимаемого уровня в команде мобильной разработке условия труда относятся к оптимальным, а степень напряженности труда – к легкой.

Для повышения устойчивости работников к эмоциональным перегрузкам существует ряд мероприятий:

1. Повышение профессионализма и квалификации;
2. Проведение тренингов;
3. Своевременное умеренное отвлечение от трудового процесса;
4. Создание условий по организации отдыха.

5.2.4. Опасность поражения электрическим током

Рабочее место инженера-разработчика в обязательном порядке оборудовано таким электрооборудованием, подключенном к сети, как системный блок и монитор. Кабинет оборудован 6 рабочими местами. Таким образом, из-за наличия большого числа электрооборудования в рабочем помещении существует опасность поражения электрическим током.

Удар электрическим током отражается серьезными последствиями на организме человека. Он может вызывать ожоги, сбой в работе сердечной мышцы или нервной системы. При поражении головы возможна потеря сознания. По статистике 8% случаев заканчиваются летальным исходом.

Рабочее помещение инженера-разработчика можно отнести к классу без повышенной опасности. Несмотря на это, потенциальное поражение электрическим током остается возможным при коротком замыкании и соприкосновении с элементами, находящимися под напряжением.

Риск поражения электрическим током зависит от влажности и температуры воздуха, а также наличия токопроводящей поверхности. Для устранения возможности данного удара в рабочем помещении поддерживается оптимальный микроклимат, а пол выполнен из не проводящего ток линолеума.

Для повышения электробезопасности необходимо соблюдение простого ряда действий:

1. Использование качественной изоляции;
2. Соблюдение сотрудниками техники безопасности;
3. Проверка работниками исправности выключателей и розеток перед началом трудовой деятельности;
4. Диагностика рабочих помещений в области электробезопасности.

5.3. Экологическая безопасность

В настоящее время экологическая проблема является одной из основных проблем человечества. Нарушение структуры и функционирование природной среды посредством влияния человека продолжает наращивать обороты. Для поддержания экологии необходимо проводить комплекс мер, с помощью которых можно снизить пагубное влияние на окружающую среду от человеческой деятельности.

Реализованное решение не несет нагрузки в усугубление экологической проблемы, так как даже не обладает физической формой. Достижение поставленной цели, в свою очередь, осуществлялось с помощью персонального компьютера, использование которого негативно отражается на окружающей среде.

При создании компьютерной техники используется большое количество химических веществ. Многие из них, такие как мышьяк, свинец, ртуть, являются токсичными. По этой причине их утилизация строго ограничена законодательством. При возможном захоронении такого вида техники происходит загрязнение литосферы, а при сжигании – атмосферы.

Для улучшения ситуации необходимо применять принцип вторичного использования, при котором неисправное оборудование не утилизируется, а пускается на конвейер для создания нового. Такой подход, во-первых, помогает избежать проблемы утилизации отходов, а во-вторых, – держать на одном уровне использование токсичных химических веществ.

Кроме этого, для контроля токсичных веществ необходимы ограничения планетарного масштаба в области их использования [24]. Это вынудит производителей компьютерной техники пересмотреть их применение, которое, как следствие, может привести к поиску и дальнейшему использованию экологически более безопасного сырья.

Осуществление данных подходов скажется благоприятно не только на окружающей среде, но также и на здоровье людей в целом, так как минимизирует отравление опасными веществами.

5.4. Безопасность в чрезвычайных ситуациях

Работа с компьютерной техникой подразумевает постоянное использование электрического тока. При несоблюдении правил электробезопасности возможна такая чрезвычайная ситуация, как пожар.

При использовании электрооборудования причиной пожара является искра, источником которой является либо короткое замыкание, обусловленное использованием неисправного оборудования, либо нагрев участка электросети вследствие больших переходных сопротивлений или перегрузок.

Чтобы не допустить возникновения данной чрезвычайной ситуации, необходима периодическая, своевременная диагностика по обнаружению неисправностей, а также соблюдение персоналом норм пожарной безопасности.

Для минимизации последствий потенциального пожара рабочие помещения должны быть оборудованы средствами пожаротушения и пожарным оборудованием, готовым к эксплуатации. Кроме того, необходим инструктаж сотрудников по плану действий в случае возникновения чрезвычайной ситуации.

Используемое рабочее помещение оборудовано датчиком дыма ИП-21210-1, порошковым огнетушителем ОП-2(з), а также планом эвакуации. Выход из помещения осуществляется в коридор, в котором расположен пожарный шкаф с необходимым инвентарем. По периметру всего бизнес-центра расположены эвакуационные знаки безопасности.

5.5. Вывод по главе

В ходе трудовой деятельности, направленной на достижение поставленной цели, нарушений по аспектам безопасности не обнаружено.

Рассмотренные правовые и организационные вопросы в области безопасности не противоречат законодательным нормам.

Возникающие вредные и опасные производственные факторы находятся в пределах нормы.

Для решения экологической проблемы компания делает все, что от нее зависит, но данная проблема должна решаться на более высоком, глобальном уровне.

Анализ безопасности в чрезвычайных ситуациях показал готовность к их потенциальному возникновению.

Заключение

В результате выполнения работы была проведена оптимизация системы навигации для решения проблемы ее гибкости в мобильном приложении «Camfrog» под ОС Android. Функциональность была проверена отделом тестирования компании, выявленные ошибки – исправлены. После этого данные изменения были внедрены в актуальную версию приложения.

На начальном этапе было проведено исследование предметной области, состоящее из ее описания с выявлением причин возникновения проблемы, обзора существующего решения с декомпозицией вариантов проявления недостаточной гибкости системы навигации, анализа альтернатив, завершившегося выбором реализации подхода с Single Activity.

После этого было выполнено проектирование, в ходе которого составлен ряд сопутствующих разработке диаграмм с использованием нотаций UML, IDEF3, DFD, BPMN и EPC.

Далее была проведена реализация оптимизации системы навигации, сопровождаемая подробным описанием. Также были приведены примеры использования новых функциональных возможностей мобильного приложения с представлением диаграммы классов.

На этапе финансового анализа были выявлены конкурентные черты разработки собственного решения, бюджет и сроки реализации.

Результат этапа социальной ответственности продемонстрировал отсутствие нарушений при выполнении выпускной квалификационной работы по различным аспектам в области безопасности.

В настоящее время изменения по модулю навигации считаются завершенными. В силу его сложности и применимости ко всем частям приложения, процесс тестирования и сопровождения продолжается.

Таким образом, в рамках выпускной квалификационной работы были закреплены навыки моделирования, проектирования и программирования.

Список достижений

Статьи:

1. Черкашин А. Ю. , Ключов П. Д. , Ворожейкин В. А. Национальный стандарт передачи данных для нефтегазовой промышленности России // Нефть и газ-2017: сборник трудов 71-ой Международной молодежной научной конференции «Нефть и газ – 2017»., Москва, 18-20 Апреля 2017. – Москва: РГУ нефти и газа имени И.М. Губкина , 2017 – С. 405-414.

Стипендии:

1. Именная стипендия ПАО «Транснефть» (г. Москва) студентам ТПУ приказ № 10181/С от 19.11.2018 года с 01 июля 2018 г. по 30 июня 2019 г.

Список источников

1. Camfrog – Википедия [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/Camfrog> (дата обращения: 04.03.2019).
2. SDK Platform release notes [Электронный ресурс] URL: <https://developer.android.com/studio/releases/platforms> (дата обращения: 04.03.2019).
3. Лицензия на вождение болида, или почему приложения должны быть Single-Activity [Электронный ресурс] URL: <https://habr.com/ru/company/redmadrobot/blog/426617/> (дата обращения: 05.03.2019).
4. Cicerone – простая навигация в Андроид приложении [Электронный ресурс] URL: <https://habr.com/ru/company/mobileup/blog/314838/> (дата обращения: 05.03.2019).
5. Разбираемся с Conductor [Электронный ресурс] URL: <https://habr.com/ru/post/329532/> (дата обращения: 05.03.2019).
6. UML – диаграмма вариантов использования (use case diagram) [Электронный ресурс] URL: <https://habr.com/ru/post/47940/> (дата обращения: 06.03.2019).
7. МЕТОД ОПИСАНИЯ ПРОЦЕССОВ IDEF3 [Электронный ресурс] URL: <http://libraryno.ru/9-metod-opisaniya-processov-idef3-trpo/> (дата обращения: 06.03.2019).
8. Что такое DFD (диаграммы потоков данных) [Электронный ресурс] URL: <https://habr.com/ru/company/trinion/blog/340064/> (дата обращения: 06.03.2019).
9. Сравнительный анализ нотаций моделирования бизнес-процессов [Электронный ресурс] URL: <https://www.osp.ru/os/2011/08/13011140/> (дата обращения: 06.03.2019).
10. Нотации моделирования бизнес-процессов [Электронный ресурс] URL: https://www.businessstudio.ru/products/business_studio/notations/ (дата обращения: 07.03.2019).

11. Нотация EPC: описание, правила построения [Электронный ресурс] URL: <http://projectimo.ru/biznes-processy/notaciya-epc.html> (дата обращения: 07.03.2019).
12. Выбор среды разработки для программирования под Android [Электронный ресурс] URL: <https://venomwind.wixsite.com/stardroid/blank-thepk> (дата обращения: 08.03.2019).
13. SWOT анализ: основы метода и ключевые элементы [Электронный ресурс] URL: <http://powerbranding.ru/biznes-analiz/swot/> (дата обращения: 12.05.2019).
14. 6 секретов планирования, или зачем необходимо составлять планы? [Электронный ресурс] URL: <https://www.b17.ru/article/555/> (дата обращения: 12.05.2019).
15. Видяев, И.Г. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение [Текст]: учебно-методическое пособие / И.Г. Видяев, Г.Н. Серикова, Н.А. Гаврикова, Н.В. Шаповалова, Л.Р. Тухватулина, З.В. Криницына; Томский политехнический университет. – Томск: Издательство Томского политехнического университета, 2014. – 36 с.
16. Производственный календарь на 2019 год [Электронный ресурс] URL: <https://spmag.ru/articles/proizvodstvennyy-kalendar-2019-s-prazdnikami-i-vyhodnymi> (дата обращения: 13.05.2019).
17. Формирование годового бюджета компании [Электронный ресурс] URL: http://www.bud-tech.ru/annual_budget.html (дата обращения: 13.05.2019).
18. Статья 58. Пониженные тарифы страховых взносов для отдельных категорий плательщиков страховых взносов в переходный период 2011 - 2027 годов [Электронный ресурс] URL: http://www.consultant.ru/document/cons_doc_LAW_89925/900cde62a67d9fc4449e78c7816ff1f56c08022f/#dst40 (дата обращения: 13.05.2019).
19. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования

[Электронный ресурс] URL: <http://docs.cntd.ru/document/1200003913> (дата обращения: 04.05.2019).

20. СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" [Электронный ресурс] URL: <http://docs.cntd.ru/document/901865498> (дата обращения: 04.05.2019).

21. «Трудовой кодекс Российской Федерации» [Электронный ресурс] URL: http://www.consultant.ru/document/cons_doc_LAW_34683/ (дата обращения: 06.05.2019).

22. ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация [Электронный ресурс] URL: <http://docs.cntd.ru/document/1200136071> (дата обращения: 07.05.2019).

23. 2.2.2006-05 «Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда» [Электронный ресурс] URL: https://ohranatruda.ru/ot_biblio/norma/246226/ (дата обращения: 07.05.2019).

24. Грязная и опасная сторона технологий [Электронный ресурс] URL: <https://www.osp.ru/pcworld/2013/06/13035804/> (дата обращения: 08.05.2019);

25. Camfrog - Group Video Chat - Revenue & Download estimates - Google Play Store – US [Электронный ресурс] URL: <https://sensortower.com/android/US/camshare-inc/app/camfrog-group-video-chat/com.camshare.camfrog.android/overview> (дата обращения: 04.03.2019).

26. Deep Linking для мобильных приложений [Электронный ресурс] URL: <https://habr.com/ru/company/redmadrobot/blog/267587/> (дата обращения: 07.03.2019).

Приложение А

Таблица 1 – Диаграмма Гантта

Дата	01.09.18	02.09.18	05.09.18	11.09.18	12.09.18	08.10.18	09.10.18	10.01.19	01.02.19	02.02.19	03.02.19	19.03.19	30.03.19	07.04.19	14.04.19	03.05.19	04.05.19	05.05.19	06.05.19	14.05.19	15.05.19	21.05.19
Продолжительность, дни	1	2	5	11	12	38	39	132	154	155	156	200	211	219	226	245	246	247	248	256	257	263
Постановка задания	1																					
Подбор и изучение литературы		1	4	10																		
Анализ предметной области			1	7																		
Перепроектирование существующего решения					1	27																
Разработка решения							1	94	116	117	118	162	173									
Тестирование												1	12	20	27							
Доработка решения														1	8	27						
Отчет о проделанной работе																	1					
																	1					
Выбор научного руководителя								1														
Утверждение темы бакалаврской работы									1													
									1													
Составление календарного плана выполнения бакалаврской работы											1	2										
Согласование основной части проделанной работы с научным руководителем																		1				
																	1	2				
Выполнение оставшихся частей																			1	9		
Подведение итогов, оформление																					1	7
	Руководитель команды мобильной разработки					Разработчик					Тестировщик					Научный руководитель						