ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

|  |
| --- |
| Институт компьютерных наук и технологий  Высшая школа программной инженерии |
|  |

**ИНДИВИДУАЛЬНЫЙ ПЛАН (ЗАДАНИЕ И ГРАФИК)**

**ПРОВЕДЕНИЯ ПРАКТИКИ**

|  |
| --- |
| Коновалов Александр Анатольевич |

|  |
| --- |
| Направление подготовки (код/наименование) 09.03.04 «Программная инженерия» |
| Профиль (код/наименование) 09.03.04\_01 «Технология разработки и сопровождения качественного программного продукта» |
| Вид практики: научно-исследовательская работа |
| Тип практики: распределенная |
| Место прохождения практики ФГАОУ ВО «СПбПУ», ИКНТ, ВШПИ, СПб, ул.Политехническая, 29 |

|  |
| --- |
| Руководитель практической подготовки от ФГАОУ ВО «СПбПУ»: |

|  |  |
| --- | --- |
| Котлярова Лина Павловна, старший преподаватель ВШПИ ИКНТ | Петров Александр Владимирович, старший преподаватель ВШПИ ИКНТ |

*(Ф.И.О., уч.степень, должность)*

|  |
| --- |
| Руководитель практической подготовки от профильной организации: - |

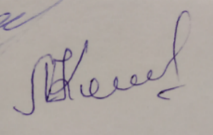
**Рабочий график проведения практики**

Сроки практики: с01.09.22 по 16.12.22

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Этапы (периоды) практики** | **Вид работ** | **Сроки прохождения этапа (периода) практики** |
| 1 | Организационный этап | Установочная лекция для разъяснения целей, задач, содержания и порядка прохождения практики, инструктаж по технике безопасности, выдача сопроводительных документов по практике | 01.09 |
| 2 | Основной  этап | Разработка веб-приложения для настройки функционала IPTV-приставок провайдером | 02.09-15.12 |
| 3 | Заключительный этап | Защита отчета по практике | 16.12 |



Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Коновалов А. А. /



Руководитель практической подготовки

от ФГАОУ ВО «СПбПУ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Котлярова Л. П. /

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

|  |
| --- |
| Институт компьютерных наук и технологий  Высшая школа программной инженерии |
|  |

**Отчет о прохождении научно-исследовательской работы**

|  |
| --- |
| Коновалов Александр Анатольевич |

|  |
| --- |
| 4 курс, 3530904/90105 |

|  |
| --- |
| 09.03.04 «Программная инженерия» |

|  |
| --- |
| **Место прохождения практики:** ФГАОУ ВО «СПбПУ», ИКНТ, ВШПИ, |

|  |
| --- |
| СПб, ул.Политехническая, 29 |

|  |
| --- |
| **Сроки практики:** 01.09.22-16.12.22 |

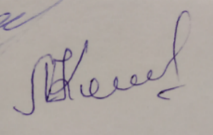
|  |
| --- |
| **Руководитель практической подготовки от ФГАОУ ВО «СПбПУ»:** |

|  |  |
| --- | --- |
| Котлярова Лина Павловна, старший преподаватель ВШПИ ИКНТ | Петров Александр Владимирович, старший преподаватель ВШПИ ИКНТ |

|  |
| --- |
| **Руководитель практической подготовки от профильной организации:** - |

|  |
| --- |
|  |

|  |
| --- |
| **Оценка: зачтено** |



|  |
| --- |
| Руководитель практической подготовки  от ФГАОУ ВО «СПбПУ»: / Котлярова Л. П. / |

|  |
| --- |
| Руководитель практической подготовки  от профильной организации: / Колычев А. В. / |

|  |
| --- |
| Обучающийся: / Коновалов А. А. / |

|  |
| --- |
| Дата: 16.12.22 |

# Оглавление

[Оглавление 3](#_Toc125395682)

[Список обозначений и сокращений 4](#_Toc125395683)

[1 Постановка задачи 5](#_Toc125395684)

[2 Обоснование актуальности работы 6](#_Toc125395685)

[3 Обзор существующих решений 7](#_Toc125395686)

[4 Обоснование выбора технологий и средств разработки 8](#_Toc125395687)

[5 Описание планируемой архитектуры 9](#_Toc125395688)

[6 Список используемых источников информации 11](#_Toc125395689)

# Список обозначений и сокращений

**Dart** – язык программирования созданный компанией Google [1]. Dart является статически типизированным объектно-ориентированным языком программирования [2].

**Flutter** – фреймворк с открытым исходным кодом, созданный Google. Он используется для разработки кроссплатформенных приложений для Android, iOS, Linux, macOS, Windows, Google Fuchsia и Web-браузеров из единой кодовой базы с помощью языка программирования Dart [3].

**IP** – маршрутизируемый протокол сетевого уровня стека TCP/IP.

**IPTV** – Internet Protocol Television (рус. Интерактивное телевидение). Технология цифрового телевидения в сетях передачи данных по протоколу IP [4].

**IPTV**-**Middleware** – промежуточное серверное ПО, позволяющее управлять всеми компонентами системы IPTV [5].

**Provisioning** – метод автоматической централизованной настройки IPTV-приставок [6].

**UI** – User Interface (рус. Пользовательский интерфейс)

**XML** – Расширяемый язык разметки [7]

**ОС** – Операционная система

**ПО** – Программное обеспечение

# Постановка задачи

Целью научно-исследовательской работы является создание веб-приложения для настройки IPTV-приставок провайдером с помощью технологии Provisioning, разработанной компанией TVIP[8]. Веб-приложение должно заменить существующий способ настройки, осуществляемый составлением XML файла с настройками, и предоставить удобный интерфейс для администраторов операторов связи.

Для достижения поставленной цели планируется разработать веб-приложение с помощью кроссплатформенного фреймворка Flutter от компании Google, созданного на основе языка Dart.

# Обоснование актуальности работы

В настоящее время существует множество способов доставки телевизионного контента от оператора цифрового телевидения до конечного пользователя, зрителя. Одним из решений является телевидение по протоколу IP (Интерактивное телевидение) (англ. Internet Protocol Television) (IPTV). В связи с развитием интернета эта технология становится все более популярной, так как обладает многими преимуществами, например воспроизведением телепрограмм на любом устройстве, удовлетворяющим системным требованиям и подключенном к сети интернет, доставка пользователю дополнительного контента и др.

Ключевым компонентом технологии IPTV является IPTV-Middleware[5], именно благодаря ему у операторов связи есть возможность управлять контентом, предоставляемым пользователям. Однако, помимо контента, операторы цифрового телевидения заинтересованы в контроле за устройствами клиентов. В контроль над устройством входит непосредственно само управление приставками. Примером которого может служить блокировка функционала приставки в случае расторжения договора об оказании услуг, это актуально, так как основная часть приставок предоставляется оператором связи вместе с услугами телевещания. В таком случае вместе с утратой основного контента оборудование пользователя лишается всех остальных возможностей. Также важной частью контроля является автоматизация настройки устройств, что облегчает работу как пользователю, так и оператору связи, и ускоряет подключение новых абонентов, получать все необходимые настройки автоматически по сети, избавляя от ручной настройки. Следующей особенностью является брендирование, позволяющее изменять пользовательский интерфейс в соответствии с общим дизайн-кодом оператора связи, приобретающего данное ПО. И, разумеется, оператору связи необходим сбор статистики с устройств пользователей, с целью улучшения качества предоставляемых услуг.

Все описанные задачи позволяет решить метод автоматической централизованной настройки функционала IPTV-приставок – Provisioning[6]. На данном этапе вся настройка производится посредствам конфигурирования XML файла с настройками, однако для такого широкого спектра задач необходим инструмент с удобным пользовательским интерфейсом, который позволит легко производить настройку администраторам операторов связи. В качестве такого инструмента выступает веб-приложение, созданию которого и посвящена выпускная квалификационная работа.

# Обзор существующих решений

Описываемый метод централизованной настройки устройств впервые был разработан и представлен компанией TVIP[8], однако позже появился его аналог от другой компании.

Так, компания ELTEX предоставляет решение «Сервер конфигурации - ACS (Automatic Configuration Server)» [9], представляющее собой веб-приложение для автоматизации и удобства настройки абонентских устройств.

Основными критериями сравнения решений будем считать: поддерживаемые платформы; функционал, доступный для настройки; наличие локализации.

В таблице 1 приведены результаты сравнения предлагаемого решения с аналогом по ряду критериев. Таблица будет дополнена данными при написании первой главы выпускной квалификационной работы.

Таблица 1 Сравнение методов централизованной настройки IPTV-приставок

|  |  |  |
| --- | --- | --- |
| Критерий | Provisioning | ACS |
| Поддерживаемые платформы | Все платформы с наличием интернет-браузера | Все платформы с наличием интернет-браузера |
| Наличие локализации | Есть | Нет |
| Возможность смены ОС приставки | Есть | Нет |
| Возможность изменения пользовательского интерфейса приставки | Есть | Нет |
| Настройка ролей и пользователей | Есть | Есть |

Обобщая, система Provisioning реализует большую часть функций ELTEX ASC, но не ограничивается на этом. Provisioning превосходит аналог, по количеству функций и глубине их проработки, поэтому в предлагаемом решении не будут заимствоваться идеи ELTEX ACS.

# Обоснование выбора технологий и средств разработки

Для разработки веб-приложения выбран кроссплатформенный фреймворк Flutter[3], основанный на языке Dart[2]. Он позволяет создавать приложения под различные платформы, в числе которых: Web (браузеры: Chrome, Firefox, Safari, Edge), Android, IOS, GNU/Linux, Windows, MacOS.

С помощью Flutter в компании уже реализовано несколько проектов, имеются собственные закрытые библиотеки и наборы средств разработки. Также в будущем будет возможность распространить созданное решение на другие платформы, сохранив слой бизнес-логики приложения и многие визуальные компоненты. По этим причинам в качестве основного инструмента разработки выбран именно Flutter.

Средой разработки выбрана AndroidStudio[10] от компании Google. Это одна из наиболее популярных сред разработки для Flutter, она удобна в использовании и позволяет запускать и тестировать приложения на всех поддерживаемых платформах.

Для контроля версиями проекта и организации работы используется платформа GitLab[11], который работает с системой контроля версий Git[12].

# Описание планируемой архитектуры

Язык Dart является статически типизированным языком программирования, использующим принцип объектно-ориентированного программирования (ООП).

Архитектурно разрабатываемое веб-приложение представляет собой 3 слоя: UI-слой (UI-layer), слой бизнес-логики (Business-logic layer) приложения и слой работы с данными (Data-layer). Такое разбиение по слоям позволяет разграничить различные части приложения и уменьшает связность компонентов, позволяя более гибко изменять их. Зависимости слоев направленны строго внутрь, UI-слой зависит от слоя бизнес-логики, который в свою очередь зависит от слоя данных.

UI-слой отвечает за ту часть приложения, которую пользователь видит на экране, все внешние компоненты (текст, кнопки и др.), этот слой получает данные из слоя бизнес-логики. Он осведомлен о формате необходимых данных , но ничего не знает об устройстве слоя бизнес-логики и общается с ним по определенному интерфейсу (контракту). При этом слой бизнес-логики, как и другие, можно подменить в любой момент, допустим тестовыми данными, главное, чтобы он удовлетворял описанному контракту. На этом слое используются стандартные инструменты фреймворка Flutter для создания UI элементов.

Слой бизнес-логики отвечает за прием и обработку данных от слоя данных, суть обмена данными такая же, как UI-слой общается со слоем бизнес-логики. Также этот слой ответственен за навигацию внутри приложения. Для этого используются инструменты Flutter, сторонние библиотеки, такие как Flutter BLoC[12], и закрытые библиотеки, разработанные силами компании, облегчающие контроль за состоянием приложения.

Самым нижним слоем выступает слой данных. На этом слое сосредоточена работа по получению данных с сервера, обработке возникающих ошибок и представлении полученных данных в виде объектов, которые передадутся ниже, на слой бизнес-логики.

На рисунке 1 представлена схема планируемой архитектуры веб-приложения. Окружности внутри блока приложения представляют описанные выше слои, стрелками показаны зависимости между ними. Также отдельным блоком выделен удаленный сервер (Remote Server), к которому приложение обращается посредствам запросов (Request), и принимает от него ответы (Response).

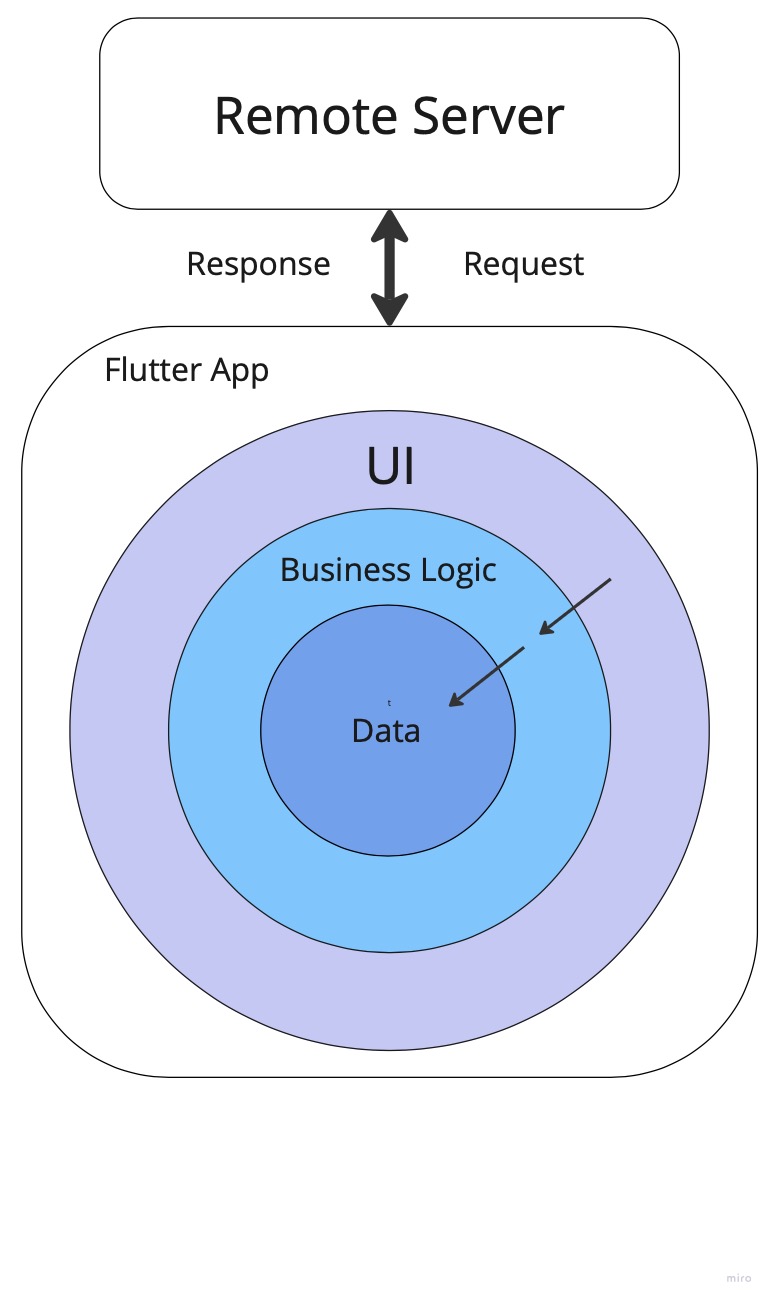


Рис. Схема архитектуры веб-приложения

# Список используемых источников информации

1. About Google [Электронный ресурс]. URL: https://about.google/ (Дата обращения: 10.12.2022).
2. Dart Language [Электронный ресурс]. URL: https://dart.dev/ (Дата обращения: 10.12.2022).
3. Flutter [Электронный ресурс]. URL: https://flutter.dev/ (Дата обращения: 10.12.2022).
4. IPTV [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/IPTV (Дата обращения: 10.12.2022).
5. Что из себя представляет IPTV Middleware? [Электронный ресурс]. URL: https://habr.com/ru/post/168891/ (Дата обращения: 10.12.2022).
6. TVIP Provisioning [Электронный ресурс]. URL: https://wiki.tvip.tv/en/provisioning (Дата обращения: 10.12.2022).
7. XML [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/XML (Дата обращения: 10.12.2022).
8. Компания TVIP [Электронный ресурс]. URL: https://www.tvip.tv/?ysclid=ld7l47v8h3574784742 (Дата обращения: 10.12.2022).
9. ELTEX.ACS [Электронный ресурс]. URL: https://eltex-co.ru/catalog/management/eltex-acs/ (Дата обращения: 10.12.2022).
10. Android Studio [Электронный ресурс]. URL: https://developer.android.com/studio (Дата обращения: 10.12.2022).
11. GitLab [Электронный ресурс]. URL: https://about.gitlab.com/ (Дата обращения: 10.12.2022).
12. Git [Электронный ресурс]. URL: https://git-scm.com/ (Дата обращения: 10.12.2022).
13. Flutter BLoC [Электронный ресурс]. URL: https://pub.dev/documentation/flutter\_bloc/latest/ (Дата обращения: 10.12.2022).