

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК



(подпись) _____ (Ф.И.О.)
« ____ » _____ 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ПРОЕКТ «МОНИТОРИНГ IT-КОНФЕРЕНЦИЙ»: ПРОЕКТИРОВАНИЕ И УПРАВЛЕНИЕ

Научный руководитель: Юманова И.Ф., руководитель
к. ф.-м. н., доцент

Нормоконтролер: Котегова А.В.

Студент группы РИМ-291219, Акулич С.А.

 подпись
 подпись
 подпись

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего
образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ
Школа профессионального и академического образования
Направление подготовки 09.04.04 Программная инженерия
Образовательная программа Разработка программно-информационных систем

УТВЕРЖДАЮ
РОП _____
« ____ » _____ 2021 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы
студента Акулича Сергея Андреевича группы РИМ-291219
(фамилия, имя, отчество)

1. Тема выпускной квалификационной работы ПРОЕКТ «МОНИТОРИНГ IT-КОНФЕРЕНЦИЙ»: ПРОЕКТИРОВАНИЕ И УПРАВЛЕНИЕ

Утверждена распоряжением по институту от « ____ » _____ 2021 г. № ____

2. Руководитель _____
(Ф.И.О., должность, ученое звание, ученая степень)

3. Исходные данные к работе Техническая литература

4. Перечень демонстрационных материалов презентация в MS PowerPoint с использованием проектора

5. Календарный план

№ п/п	Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
1.	<i>Анализ предметной области</i>	до 28 февраля 2021 г.	
2.	<i>Анализ методологий и средств управления проектами</i>	до 28 марта 2021 г.	
3.	<i>Реализация дизайн системы</i>	до 18 апреля 2021 г.	
4.	<i>Проектирование системы</i>	до 02 мая 2021 г.	
5.	<i>Разработка моделей коммерческого использования системы</i>	до 15 мая 2021 г.	
6.	<i>Оформление ВКР</i>	до 30 мая 2021 г.	

Руководитель _____
(подпись)

Куликов И.П.
Ф.И.О.

Задание принял к исполнению _____
дата

_____ (подпись)

6. Выпускная квалификационная работа закончена «31» мая 2021г. считаю возможным допустить Акулича Сергея Андреевича к защите выпускной квалификационной работы закончена в Государственной экзаменационной комиссии.

Руководитель _____
(подпись)

Куликов И.П.
Ф.И.О.

7. Допустить Акулича Сергея Андреевича к защите магистерской диссертации в Государственной экзаменационной комиссии.
РОП _____

(подпись)

_____ Ф.И.О.

РЕФЕРАТ

Выпускная квалификационная работа магистра 74 с., 29 рис., 13 табл., 29 источников, 4 прил.

**СЕРВИС, ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ,
КОНЦЕПЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ, УПРАВЛЕНИЕ В
КОМАНДЕ РАЗРАБОТКИ, ПРОТОТИПИРОВАНИЕ СИСТЕМЫ**

Цель работы – Проектирование разрабатываемого сервиса и управление им.

Объект исследования – проектирование информационной системы.

Методы исследования: анализ существующих решений, методы активации творческого мышления, метод экспертных оценок.

Результаты работы: разработана проектная документация и прототипы сервиса «Мониторинг IT-конференций».

СОДЕРЖАНИЕ

РЕФЕРАТ	3
СОДЕРЖАНИЕ	4
ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	5
ВВЕДЕНИЕ	8
1 Методологии управления проектами	12
2 Средства управления проектами	24
3 Формирование проекта «Мониторинг IT конференций»	36
4 Прототипы сервиса	39
5 Проектирование системы	49
6 Экономическое обоснование	61
ЗАКЛЮЧЕНИЕ	65
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	67
ПРИЛОЖЕНИЕ А	71
ПРИЛОЖЕНИЕ Б	72
ПРИЛОЖЕНИЕ В	73
ПРИЛОЖЕНИЕ Г	74

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете по ВКР применяют следующие сокращения и обозначения:

Backlog – это упорядоченный набор элементов, очередь задач, перечень всех функций, которые заинтересованные люди хотят получить от продукта. Этот список содержит краткие описания всех желаемых возможностей продукта.

Sprint – промежуток времени, достаточный для выполнения запланированной совокупности задач команды разработки, целью которой является создание инкремент бизнес-продукта.

Style guide – это специально разработанная система тщательно подобранных и скомпонованных элементов и правил. Style guide необходимо использовать для формирования единого стиля и целостного дизайна как разработчикам, так и дизайнерам при создании сайта, а также его отдельных страниц и компонентов.

BPMN (Business Process Management Notation) – Нотация моделирования бизнес-процессов, это метод составления блок-схем, отображающий этапы выполнения бизнес-процесса от начала до конца. BPMN-схемы наглядно и подробно демонстрируют последовательность рабочих действий и перемещение информационных потоков, необходимых для выполнения процесса, а потому являются одним из ключевых инструментов управления бизнесом.

ГБ – Гигабайт

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

HTTP (HyperText Transfer Protocol) – протокол прикладного уровня передачи данных, изначально – в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.

ER-модель – модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Backend – это программно-аппаратная часть сервиса. Это набор средств, с помощью которых происходит реализация логики вебсайта.

Frontend – клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса. Это набор средств, с помощью которых происходит визуализация данных от программного-аппаратной части сервиса.

DevOps – это набор практик для повышения эффективности процессов разработки (Development) и эксплуатации (Operation) программного обеспечения (ПО) за счет их непрерывной интеграции и активного взаимодействия профильных специалистов с помощью инструментов автоматизации.

UI (User Interface, Пользовательский интерфейс) – это визуальный результат работы дизайнера, то, что видит пользователь. UI-дизайнер придает материальную форму архитектуре пользовательского опыта и «оживляет» интерфейс продукта. Интерфейс создается на основе исследований пользовательского опыта – UX.

Snackbar – это маленький компонент интерфейса, предоставляющий короткий ответ после действия пользователя, он появляется в самом конце экрана и автоматически исчезает после того, как время истекло или когда пользователь взаимодействует в другом месте экрана.

Django – свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования Model-View-Controller.

JSON Web Token – это открытый стандарт для создания токенов доступа, основанный на формате JSON, который используется для передачи данных для аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности.

API (Application programming interface) – это набор готовых классов, функций, процедур, структур и констант. Вся эта информация предоставляется самим приложением (или операционной системой). При этом пользователю не обязательно понимать, что это API технология обеспечивает взаимодействие модулей. Цель представленной информации – использование этих данных при взаимодействии с внешними программами.

Kubernetes – это платформа с открытым исходным кодом, которая автоматизирует операции с контейнерами. Она упрощает многие ручные процессы, связанные с развертыванием и масштабированием упакованных в контейнеры приложений. Таким образом, можно объединять группы хостов, работающих под управлением контейнеров Linux, и Kubernetes поможет эффективно управлять этими кластерами, которые могут охватывать узлы в общедоступных, частных или гибридных облаках.

Фреймворк – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

ВВЕДЕНИЕ

Актуальность темы исследования. На рынке предоставляющем информацию о работе новых технологий, таких как Docker, Kubernetes, новые паттерны программирования и другой образовательный контент, присутствуют сервисы для обучения этим технологиям. Такие сервисы не создают новую информацию, а являются агрегирующими сервисами, которые структурируют уже имеющуюся в интернете информацию, что позволяет проще воспринять и усвоить информацию. Например, сервис Udemu [1] позволяет пользователям создать видео-лекции по объяснению новых технологий в сферах разработки ПО, ведению бухгалтерии, маркетингу, дизайну и других направлениях, например курс по работе с транзакциями в SQL, также сервис дает возможность монетизировать реализованной пользователем за структурированный курс.

По данным отчёта цифровой среды Global Digital за 2021 год [2], число интернет пользователей растёт. В крупные социальные сети и на образовательные порталы приходят всё больше новых клиентов, так данные исследования vc/media [3], показывает, что уже в 2020 году люди оплачивали подписки на музыку, кино, книги наравне с коммунальными услугами и питанием.

Несмотря на присутствие на рынке огромного количества обучающих сервисов, таких как Skyeng, Udemu, Skillbox и других, ни в одной из них не реализовала сбор информации с прошедших конференций по IT-технологиям, а именно: презентаций, текстов докладов, видео. По данным ресурса it-events [4], ежегодно в России проводится свыше 100 конференций, на которых рассматривают новые подходы к разработке, обсуждают проблемы и преимущества новых версий языков программирования, внедрение новых технологий, которые появляются в сообщества, например лекция от компании Яндекс о технической реализации виртуального ассистента “Как устроена Алиса?” [5].

Пользователям приходится искать необходимую информацию (запись лекции, материалы) на сайтах конференций и мероприятий. Однако это не всегда удается, так как организаторы конференций могут размещать видео не на сайтах, а на YouTube каналах. В таком случае пользователям приходится искать видео в сети Интернет самостоятельно. Это легко сделать, если пользователь знает название доклада и спикера, но появляются проблемы, когда интерес пользователя предоставляет определенное направление, например, “всё про Python”, с прошедшей конференции.

Проект «Мониторинг IT конференций» предназначен для сотрудников и работодателей компаний для поиска материалов конференций, а также для пользователей в сфере IT, которые следят за выступлениями и докладами коллег.

Каждая крупная компания, которая организует только свои лекции и мастер классы, реализует или пытается реализовать свой сервис. Доклады спикеров, которые появляются на IT-конференциях, либо загружаются на YouTube, либо материалы пропадают с онлайн-площадок.

Анализ актуальности обусловили выбор темы исследования: «Исследование и применение современных технологий для реализации технической документации и прототипов сервиса «Мониторинг IT конференций»».

Гипотеза исследования: Применение современных технологий построений бизнес-процессов и разработанных подходов их реализации в сервисе «Мониторинг IT конференций» ускорит процесс взаимодействия между IT рынком по передаче полезной информации и использованием высокотехнологичных решений на предприятиях, повысит удобство использования услуг онлайн конференций, позволит увеличить прибыли организаторов конференций на онлайн площадках.

Целью исследования является выявление и реализация оптимального способа создания сервиса «Мониторинг IT конференций», содержащего

пользовательский интерфейс приложения для быстрого поиска видео определенной направленности по заданной тематике в сфере IT.

Для достижения поставленной цели необходимо решить следующие задачи:

- анализ существующих решений в области сервисов по предоставлению информации с прошедших докладов;
- составление диаграмм бизнес-процессов сервиса в нотации Business Process Management Notation (BPMN) и обоснование ее эффективной применимости;
- выбор средств сбора, документирования, согласования и модификации требований к информационной системе (ИС);
- разработка, дизайн прототипов приложения для демонстрации потенциальным потребителям;
- оценка результатов запросов потребности сервиса на рынке.

Объектом исследования является класс сервисов для хранения, отображения и категоризации потокового видео.

Предметом исследования является техническое описание сервиса для хранения и предоставления потоковой информации, средствами Unified Modeling Language (UML) и описанием HTTP запросов взаимодействия клиент-серверной части приложения.

Методы исследования включают в себя

- анализ, сравнение, систематизация и обобщение данных о существующих и разработанных способах предоставления категоризованного контента;
- апробация современных технологий при построении веб-приложения;
- разработка дизайн-прототипа сервиса;
- анализ технологий и подходов, позволяющих масштабировать и оптимизировать WEB-приложение.

Теоретической основой исследования стали:

- зарубежные исследования и решения по организации автоматизации хранения и предоставления потоковой информации [6,7,8];
- современные концепции и технологии разработки веб-приложений;
- документация к различным используемым в современных сервисах бизнес-процессов и подходов к дизайн составляющей.

Научная новизна и теоретическая значимость исследования. Работа продолжает исследования в области управления процессами в стартапах для получения потокового видео, подходов к систематизации бизнес-процессов и структурному описанию клиент-серверной части программного обеспечения.

Выявлены, обоснованы и описаны преимущества информационных технологий как инструмента хранения и предоставления полезной информации. В работе используются нотации Business Process Management Notation и Unified Modeling Language (UML), которые показывают, что с помощью реализованного сервиса появится возможность повысить удобство нахождение материалов в сети Интернет.

Практическая значимость исследования. Работа рассматривает средства и способы управления программными проектами в стартап-компаниях и нацелена на увеличении производительности команды.

Получены результаты анализа поиска видео по заданной тематике в нотации UML при помощи диаграмм последовательности, реализованы описания взаимодействия пользователя с системой, создан рабочий дизайн проекта, реализована техническая документация обработки HTTP запросов между клиент-серверной частью приложения, структура хранения данных, с использованием базы данных PostgreSQL, предоставлены результаты опроса потенциальных потребителей.

1 Методологии управления проектами

С 1994 года компания The Standish Group ежегодно собирает статистику и предоставляет отчет «CHAOS report» [9], который отражает зависимость успешности разработки программных проектов от объема проекта и выбранной методологии разработки, где объем равен времени разработки проекта умноженный на стоимость проекта умноженный на качество проекта. Для реализации успешного проекта необходимо учитывать данные исследования «CHAOS report». Проект «Мониторинг IT-конференций» относится к группе «small» по категоризации The Standish Group, так как не имеет установленных сроков исполнения. Проект является законченным в тот момент, когда выполнены все итерации для минимальной жизнеспособности продукта (MVP). По статистике «CHAOS report», представленной в таблице 1, 11% маленьких проектов являются провальными.

Таблица 1 - Успешность программных проектов с 2011г. по 2015г.

	Успех	Переменный успех	Провал
Огромная	2%	7%	17%
Большая	6%	17%	24%
Средняя	9%	26%	31%
Небольшая	21%	32%	17%
Маленькая	62%	16%	11%
Итоги	100%	100%	100%

В таблице 2 представлена зависимость успешности проекта от выбранной методологии ведения проекта. The Standish Group указывает на важность выбора методологии на стадии создания проекта.

Таблица 2 – Зависимость успешности проекта от методологии.

Размер	Метод управления	Успех	Переменный успех	Провал
По всем проектам	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Крупные проекты	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Средние проекты	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Маленькие проекты	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

В исследовании представлена методология управления проектами каскадным методом «Водопад» или «Waterfall» [10]. Методология каскадного управления проектом предполагает сбор всех требований об информационной системе на начальном этапе разработки проекта. На рисунке 1 представлена схема разработки информационных систем каскадным методом. На этапе 1 «Анализ» происходит составление первичной документации, сбор бизнес-требований к информационной системе [11]. На данном этапе часто проводятся совещания, на которых определяется что же должен представлять из себя продукт проекта. Первый этап является ключевым, в случае ошибки в составлении документов, проект будет завершён неуспешно или частично успешно, так как будут увеличены сроки сдачи проекта. На этапе 2 «Проектирование» осуществляются распределение работ по срокам, на

основании бизнес-требований составляется техническая документация к разрабатываемому проекту, формирует календарный план и бюджет, оценивает риски и выявляет заинтересованные стороны. Этап 3 - разработка проекта на основании документации, определяется конфигурация будущего проекта и технические способы его достижения. На 4 этапе проект проверяется на соответствие требованиям, выявляются и исправляются недостатки. На этапе 5 происходит внедрение информационной системы. Схема, представленная на рисунке 3, показывает, что на любом этапе реализации проекта, есть вероятность вернуться к пункту 1 «Анализ», что влечет за собой дополнительные расходы, в связи с увеличением срока разработки проекта, в то же время увеличивается качество, реализуемого проекта, ввиду возможности возврата к изначальному состоянию.

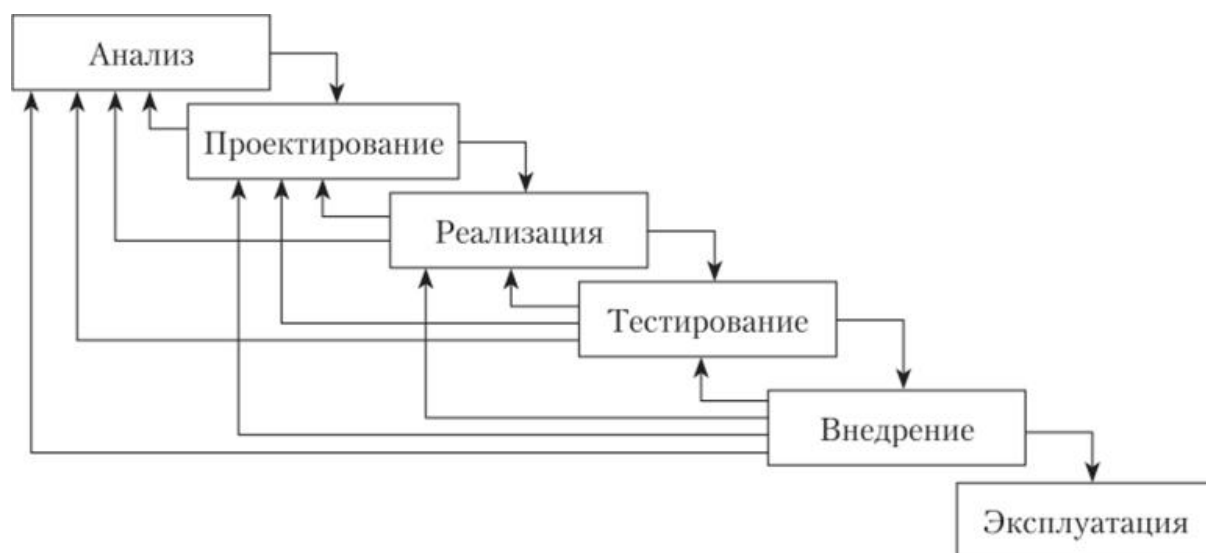


Рисунок 1 - Жизненный цикл по методологии «Водопад»

Каскадный метод разработки подходит для фундаментальных проектов, в которых качество разработки является важнее сроков реализации проекта. Данный подход ориентирован на проекты, в которых есть строгие ограничения по последовательности выполнения задач. Основным преимуществом Водопада является оценка сроков и бюджета.

В исследовании также рассматривается Agile – это семейство гибких итеративно-инкрементальных методов к управлению проектами и продуктами. Согласно данному подходу, проект разбивается не на последовательные фазы, а на маленькие подпроекты, которые затем объединяются в готовый продукт. На рисунке 2 представлена подход к разработке по Agile[12]. Agile – не является ни методологией, ни стандартом, это набор принципов и ценностей. [13]



Рисунок 2 - Принцип управления проектом по Agile

Основные принципы Agile подхода состоят в следующем:

- 1) Проект делиться на итерации, sprint - равные промежутки времени, через которые будет сохранять изменения проекта, например раз в две недели.
- 2) Формирование backlog продукта. Большие задачи, которые необходимо разработать разбиваются на маленькие задачи, которые возможно выполнить в sprint, формируя backlog продукта.

3) Каждая итерация включает в себя полный цикл разработки программного обеспечения: анализ, дизайн, разработка, тестирование, внедрение.

Преимуществом использования данного подхода определяется строгим определением требований и проверкой разрабатываемой функциональности на каждой итерации проекта.

Недостатком является необходимость самостоятельно составлять свою систему управления внутри команды, руководствуясь принципами Agile. [14]

Одним из популярных методологий использования принципов Agile в проектной разработке является Scrum [15]. Данный подход объединяет в себе элементы каскадного метода разработки с гибким подходом управления проектами. На рисунке 3 представлен процесс ведения задач команды по Scrum.



Рисунок 3 - Процесс разработки по Scrum

В данной методологии ключевую роль играет владелец продукта - участник команды, который управляет процессом разработки и несет ответственность за результат работы команды. Владелец продукта в начале

каждой итерации определяет необходимые для реализации задачи из backlog, планирует sprint, приоритизируя потребности в проекте, осуществляет контроль за выполнение во время каждой итерации, осуществляет коммуникации с командой и заказчиком.

Метод разработки по Scrum включает в себя ежедневное совещание, на котором все участники команды отвечают на три вопроса: Что делал? Что планирую делать? Существуют ли проблемы из-за которых задача может быть не выполнена в срок?

Каждая команда выбирает оптимальное время итераций, необходимое для прохождения полного жизненного цикла разработки программного обеспечения. По Scrum sprint длиться не менее одной и не более четырех недель, согласуется с заказчиком проектом.

В Scrum команде присутствует scrum-мастер - это человек, который поддерживает в команде культуру scrum и следит за соблюдением ее принципов. Этот человек во всем следует ценностям и методикам scrum, при этом действует гибко и всегда готов использовать новые возможности на благо рабочего процесса команды [16].

В обязанности scrum-мастера входит

- 1) проведение ежедневных собраний - стендапов;
- 2) собрания по планированию sprint и итераций;
- 3) обзор итогов sprint. Сбор обратной связи внутри команды, с целью улучшения текущих процессов;
- 4) ретроспективы - собрания, на которых проводится разбор слабых и сильных сторон предыдущей итерации с целью улучшения процессов;
- 5) администрирование актуального состояния этапов разработки, например, если команда использует средства по отслеживанию задач (таск-трекеры). Scrum-мастер отвечает за актуализацию текущих задач из backlog;

- 6) встречи один на один с участниками команды для развития коммуникативных навыков внутри команды, урегулирования конфликтов;
- 7) создания условий для работы команды, начиная от починки компьютера до взаимодействия между членами команды

Важно отметить, что разработка по Scrum обязывает к ролям внутри команды. Так scrum-мастер занимается взаимодействием внутри команды, урегулирование конфликтов, регулированием процессов и правильностью их движения относительно scrum методологии, в то время как владелец продукта отвечает за сроки проекта и качество реализованного функционала.

Преимуществами использования данного метода является:

- 1) тесная коммуникация в команде, которая позволяет быстрее адаптироваться в проекте новому сотруднику;
- 2) срез по функционалу, возможность показать заказчику результаты проекта после каждой итерации, что в свою очередь отражает эффективность работы.

Недостатками использования данного метода является:

- 1) тесное взаимодействие всех участников команды, подойдёт не каждой группе;
- 2) кроссфункциональность - каждый член команды должен обладать более, чем одной компетенцией, чтобы часть команды не «простаивала» на разных этапах проекта, а также для того, чтобы сотрудники могли помогать и подменять друг друга;
- 3) подходит только для разработки программного обеспечения;
- 4) много временных затрат на коммуникации и атрибуты методологии: ретроспективы, встречи один на один, планирование sprint всей командой;
- 5) отсутствуют оценки сроков и бюджета проекта.

Метод Scrum определяет, как реализовать крупный проект с помощью backlog, но не уделяет особого внимания процессам внутри итерации. Для этого существует метод Lean[17,18], который дополняет Scrum добавляя определение потока операций «workflow». Если вернуться к каскадному методу разработки, можно заметить, что весь функционал разрабатывается по определенным этапам: анализ, планирование, разработка, тестирование, внедрение. В Lean, как и в Scrum крупная задача декомпозируется на небольшие пакеты поставки, где при реализации всех пакетов будет реализована основная задача. На рисунке 4 представлен пример использования метода Lean, как и в каскадном программировании выделяются основные этапы, каждый этап разбивается на шаги для достижения основной цели этапа, после завершения всех шагов, этап считается выполненным.



Рисунок 4 - этапы в методе Lean

Отличительной особенностью Lean выделяют формирование этапов производственного цикла внутри одного sprint. Если вернуться к рисунку 5 и обратить внимание на sprint, можно заметить, что процессы внутри одной итерации не формализованы. Lean дополняет эту диаграмму, добавляя внутри каждой итерации этапы реализации, аналогично каскадному методу, одна не

строго типизирует этапы разработки. Если каскадная модель всегда содержит анализ и планирование, Lean указывает на необходимость использования этапов для повышения качества продукта, вне зависимости от самих этапов, например: в стартапах часто отсутствует заказчик, что позволяет объединить этап Анализ и Проектирование, так как отсутствуют внешние факторы влияния на проект и на проектирование системы тратиться меньше времени. Важно помнить, что Lean может являться дополнением к Scrum в общей архитектуре Agile подхода к управлению проектами.

Преимуществом использования Lean выделяют увеличение качества разработки за счет выстроенных этапов.

Недостатками в методе являются: отсутствие четкого рабочего процесса для реализации декомпозированных задач, не все задачи подходят под один и тот же описанный процесс, что увеличивает время на разработку мелкой задачи.

Для построения собственной системы управления проектами, Lean является абстрактным инструментом, для визуализации и конкретизации применяют метод Kanban [19].

Kanban позволяет сосредоточиться на разработке проекта без дополнительной роли scrum-мастера в отличие от метода Scrum, что переносит обязанность самоорганизации на команду разработки. Разница между методологиями Scrum и Kanban представлена в таблице 3.

Таблица 3 - разница методологий Kanban и Scrum

Kanban	Scrum
Нет совещаний	Есть совещания
Нужна отправная точка	Не требуется отправная точка

Продолжение таблицы 3

Могут работать узкопрофильные специалисты	Только кроссфункциональная команда
Последовательные и плавные перемены	Кардинальные перемены
В команде нет разделения на роли	Есть разделение на роли

Основные принципы Kanban:

- 1) система карточек - для каждой задачи создается индивидуальная карточка, в которую заносится вся необходимая информация о задаче. Таким образом, вся нужная информация о задаче всегда под рукой;
- 2) ограничение количества задач в sprint. Благодаря такому принципу легко отследить, на каком этапе возникают проблемы, какие процессы нужно улучшать;
- 3) непрерывный поток. Все декомпозированные задачи хранятся в списке to-do представленным на рисунке 5, который представляет backlog проекта. После реализации задачи, по приоритету из backlog берётся следующая задача;
- 4) постоянное улучшение. Основная идея в том, чтобы анализировать процесс за счет Kanban-доски, находить медленные производственные этапы и находить способы повышения эффективности;

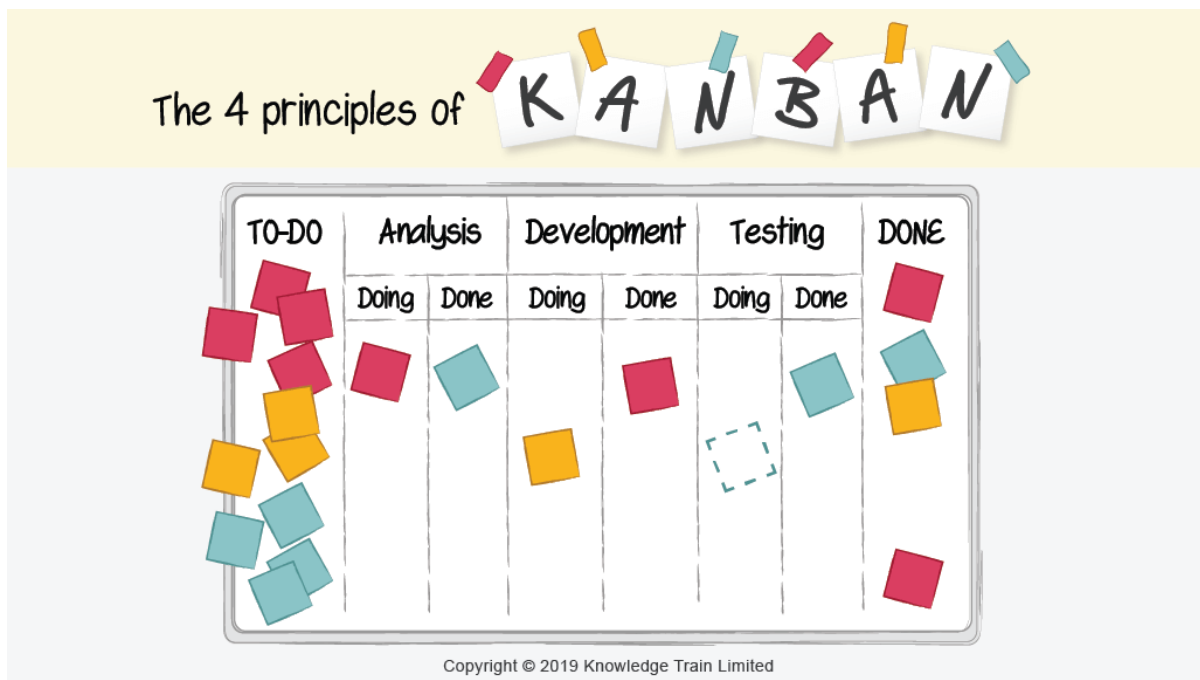


Рисунок 5 - доска Kanban

Преимуществом использования Kanban:

- 1) в Kanban нет установленных сроков исполнения задачи, они заменены приоритетами, подходит для стартап-проектов без заказчиков;
- 2) расчет нагрузки на команду и улучшение процессов, в отличие от Scrum, где в sprint берется определенное количество декомпозированных задач, в Kanban при помощи визуальной доски можно отследить, на каком этапе проекта возникают проблемы, команда самоорганизуется и вовлекается в процесс увеличения эффективности.

Недостатки использования Kanban:

- 1) Kanban лучше всего подходит для команд, навыки членов которых пересекаются друг с другом;
- 2) методология не применима при наличии в проекте установленных с заказчиком сроков реализации.

Выводы к главе 1

На основании проведенного исследования методологий можно сделать следующие выводы. Гибкая модель является наиболее подходящей для IT-компаний, стартапов, проектов в инновационных сферах, модель Kanban позволяет полноценно управлять деятельностью команды, в то время как каскадная модель удобна в проектах, где ключевым ограничителем является срок реализации проекта, а не финансы.

Для небольшой команды разработки в стартапе без жестких сроков для успешного завершения проекта, оптимальным решением будет использование Agile подхода с методологией Kanban, где используется система ведения проекта при помощи приоритизации задач с использованием визуальных карточек, для определения статуса разработки проекта.

2 Средства управления проектами

Для реализации проекта, необходимо определиться с инструментом, который будет соответствовать следующим критериям:

- 1) интуитивно-понятный интерфейс;
- 2) возможность вставки блоков кода для документирования и описания процессов;
- 3) добавление медиа-контента: ссылки, картинки, диаграммы;
- 4) создание гибкой доски по методологии Kanban с возможностью выбора исполнителя, приоритета и платформы при составлении технического задания;
- 5) не требуется плата за использование команде из трех человек;
- 6) возможность совместной работы с документами;
- 7) возможность оффлайн доступа.

Определив критерии нужного инструмента, рассмотрим существующие системы ведения задач.

Одним из популярных средств ведения проектов является информационная система Asana [20] - это универсальное решение для управления проектами и задачами, которое позволяет решать такие задачи как коммуникация внутри команды и сотрудничество с внешними партнерами.

На рисунке 6 представлен интерфейс сервиса Asana на вкладке с текущими задачами проекта, в интерфейсе представлена группировка задач по темам, есть возможность выбрать исполнителя, поставить приоритет задачи и предполагаемую дату завершения задачи.

Task name	Assignee	Due date	Team	
▼ Business				
✓ Customer base grows by 15% 3		Sep 8	Sales	
✓ Customer churn decreases by 8% 2 5		Sep 12	Support	
▼ Internal				
✓ Open NYC office		Sep 16	Facilities	
✓ Hire 18 new team members 8		Sep 6	Recruiting	
▼ Product				
✓ Increase usability with app refresh 3		Sep 14	Mobile	
✓ Secure new integration partnerships		Aug 30	Platform	
▼ Company				
✓ Raise Series B financing 7		Sep 1	Executive	

Рисунок 6 - Asana. Вкладка «Задачи»

На рисунке 7 представлен интерфейс создание задачи в Asana. Перегруженный интерфейс при создании задачи является сложным для восприятия. Компания разработала специальное руководство по ведению и управлению проектами внутри системы [21]

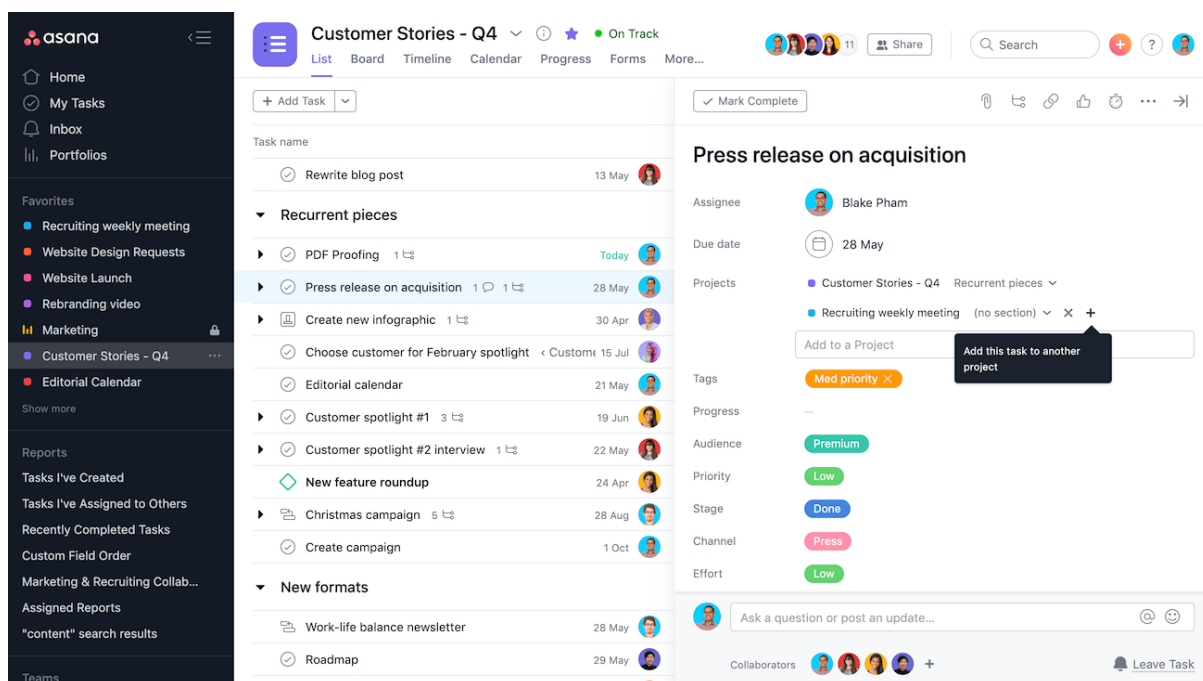


Рисунок 7 - Asana. Создание задачи

На рисунке 8 представлен интерфейс Kanban доски, отражающий текущие задачи в проекте. Важным элементом является отображение медиафайлов с развёртывание на доску, что является недостатком при составлении задач на разработку вёрстки по макетам.

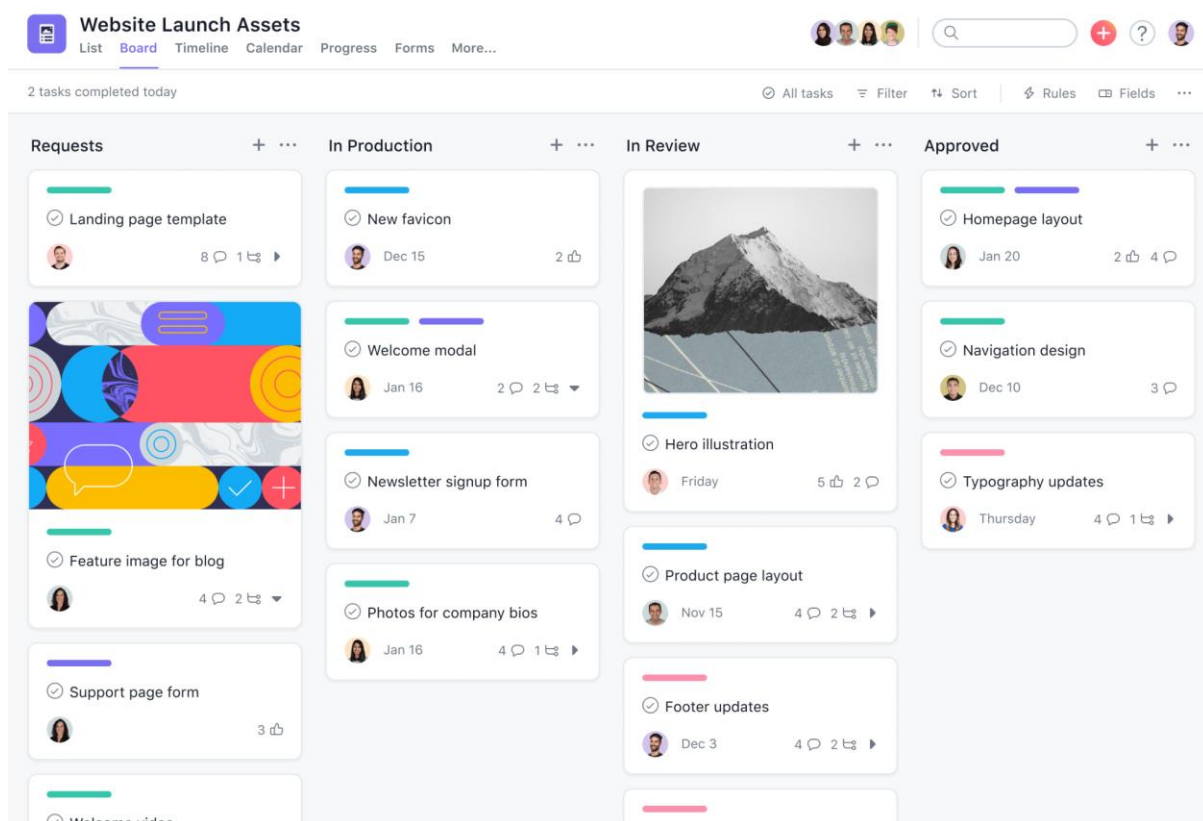


Рисунок 8 - Asana. Kanban доска

Преимущества использования Asana:

- 1) возможность пользоваться сервисом бесплатно команде до 15 человек;
- 2) визуальная приоритезация задач благодаря цветовому выделению важных задач;
- 3) возможность строить диаграммы Ганта [22];
- 4) к одной задаче можно добавить более одного тега - обозначения для быстрого поиска, например: платформа backend/frontend;
- 5) поддержка управления горячими клавишами;
- 6) добавление новых задачи без необходимости заполнения многих полей;
- 7) интеграция с электронной почтой.

Недостатки использования Asana:

- 1) отсутствует десктопная версия - сервис используется только в онлайн;
- 2) большое количество функций, перегруженный интерфейс, что создает высокий порог вхождения в программное обеспечение, необходимость изучения документации по сервису для возможности использования всех частей;
- 3) есть возможность потерять связь при перемещении элементов; чрезмерное количество функций, которые отображаются на экране, затрудняют и замедляют процесс обучения.

Для работы с данным инструментом необходимо изучить документацию по нему ввиду отсутствия интуитивно понятного интерфейса. Asana подойдет для команд и компаний, которые ведут параллельную разработку нескольких проектов благодаря возможности смены фильтрации на доске проекта, а также разграничению проектов внутри вкладки Задачи.

Популярным инструментом в промышленной разработке является Jira [23], разработанное компанией Atlassian, которая также реализовала систему ведения документации проекта Confluence.

На рисунке 9 представлена доска управления проектом в Jira. Простой интерфейс позволяет легко адаптироваться в системе. Система тегов визуально разграничивает задачи для пользователя, позволяет быстрее найти необходимую карточку на доске.

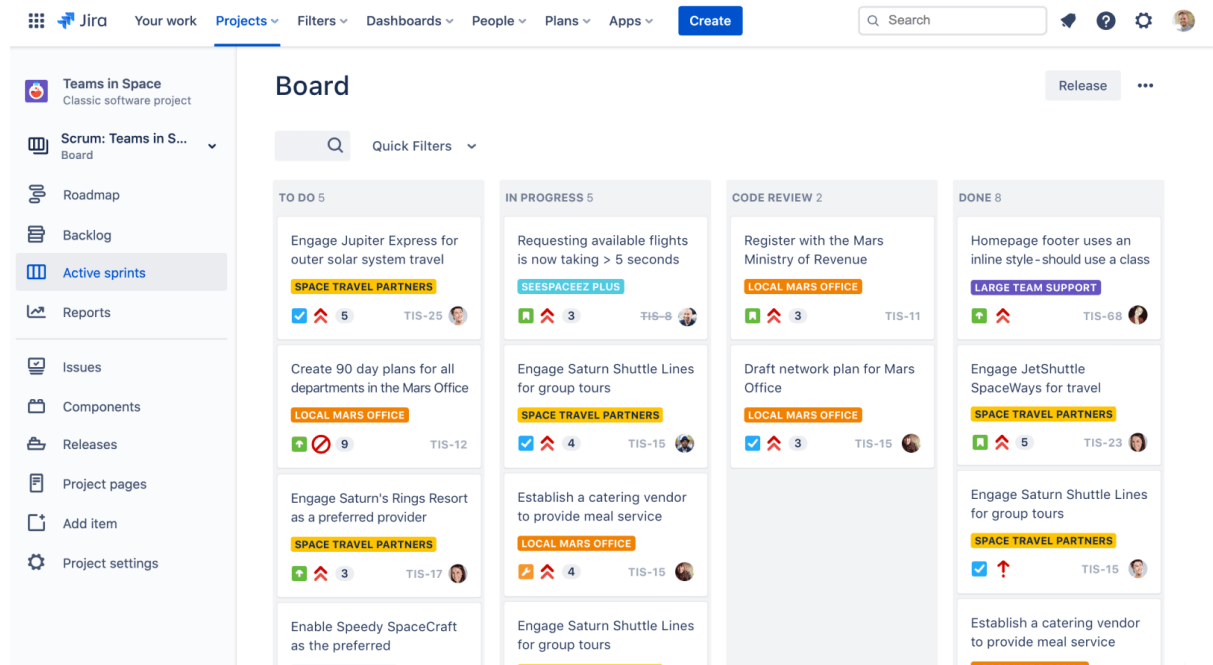


Рисунок 9 - Jira. Kanban доска

В левом боковом меню располагается backlog проекта, также можно перемещаться между досками при наличии нескольких проектов, что позволяет разделять проектирование функционала внутри системы.

На рисунке 10 представлена диаграмма Ганта проекта в системе Jira, которая позволяет отслеживать текущий статус проекта.

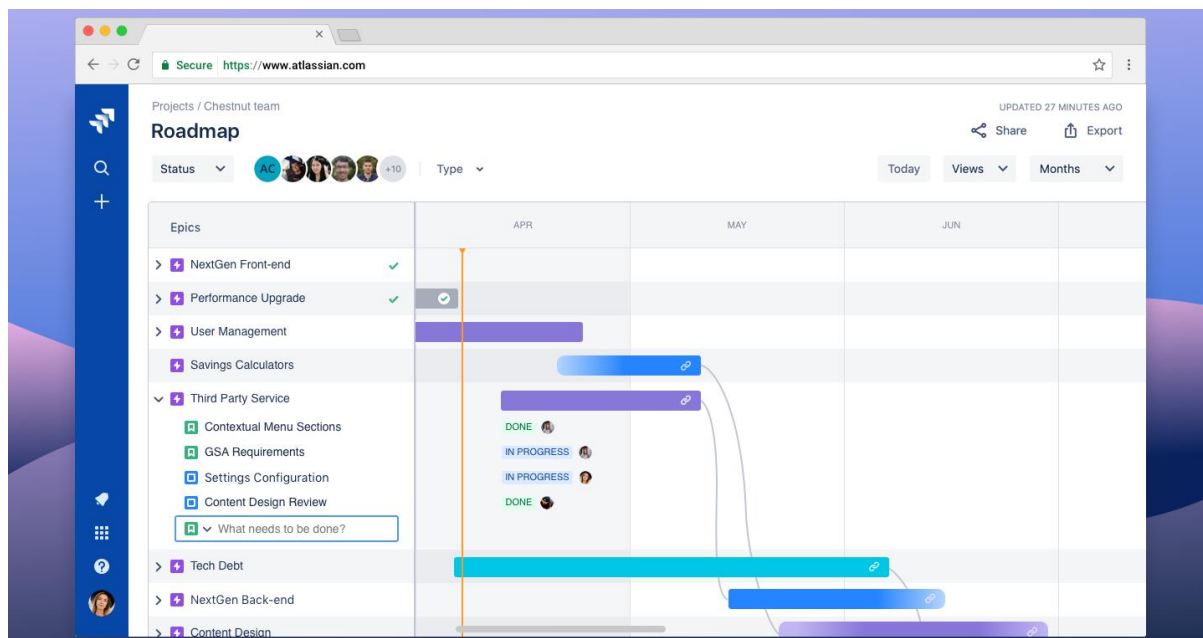


Рисунок 10 - Jira. Диаграмма Ганта

Преимущества использования Jira:

- 1) создание прямой, понятной иерархии крупной задачи является основным фактором в любой группе декомпозированных задач. В Jira есть возможность разграничения доступов членов команды к доскам проекта, что позволяет скрывать лишнюю информацию для конкретного пользователя системы, а также разделить информацию по разным доступам, создав ограничение по проекту или функционалу;
- 2) приоритезация задач дает определение всем участниками команды какие задачи необходимо выполнить немедленно, а какие могут быть решены позже. Если проект имеет четкие сроки полезно иметь функционал при работе над различными проектами с разными датами завершения, такая возможность увеличивает производительность;
- 3) backlog, где все члены команды получают возможность в любой момент времени видеть последовательность задач в списке, вследствие чего уменьшается время простоя, потраченное на

обсуждение текущих проблем, и повышается производительность работы. Внутри команды появляется понимание следующих итераций и возможных доработок системы;

- 4) более 1000 плагинов. JIRA поставляется с более чем 1000 приложениями, которые помогут сделать программное обеспечение еще более полезным для команд гибкой разработки.

Рассмотрим недостатки использования JIRA

- 1) большая насыщенность функциональными возможностями, что создает дополнительные трудности не только для начинающих пользователей, но и для продвинутых, так как процесс кастомизации продукта оказывается достаточно трудоемким в силу большого количества настраиваемых параметров системы. Таким образом, требуется некоторое время, чтобы понять архитектуру программного обеспечения;
- 2) не интуитивно понятный интерфейс: создание доступов, распределение ролей, объединение людей в группы требует изучение документации Jira;
- 3) требование наличия навыков DevOps для усиления процессов непрерывной интеграции, так как Jira рекомендуют интегрировать со средством совместной разработки github;
- 4) отсутствие встроенного вики для ведение документации, процесс документирования вынесен в отдельный сервис Confluence.

За счет большого количество функционала Jira представляет мощный агрегатор проектов, который позволит любому продакт-менеджеру отслеживать все стадии и сроки ведения одного или нескольких проектов. Инструмент является гиперболизированным для маленьких компаний, так как имеет широкий функционал. Atlassian вынесли вики систему отдельно, соответственно, для быстрого старта стартапу необходимо разбираться в двух

системах, что приводит к затратам времени на организационные моменты, и отнимает силы от реализации проекта.

В последние годы набирает популярность средство для ведения документации Notion [24] - конфигурируемая платформа для ведения рабочего пространства с возможностью работы в бесплатном режиме до 5 пользователей. Платформа ежегодно расширяется, создает новые интеграции и разрабатывает свое API для автоматизации рабочих процессов.

При создании проекта в Notion, предлагается выбор из шаблона, на что направлен проект, на выбор предоставляется: проектное управление, инжиниринг, продажи, пространство для работы кадрового отдела и другие. Среда работает по принципу вики страниц, где каждая страница содержит полезную информацию по проекту, так на рисунке 11 в верхней части экрана присутствует описание Kanban доски с основными принципами и руководствам по использованию, что позволяет пользователю быстрее разобраться в интерфейсе, не открывая документацию.

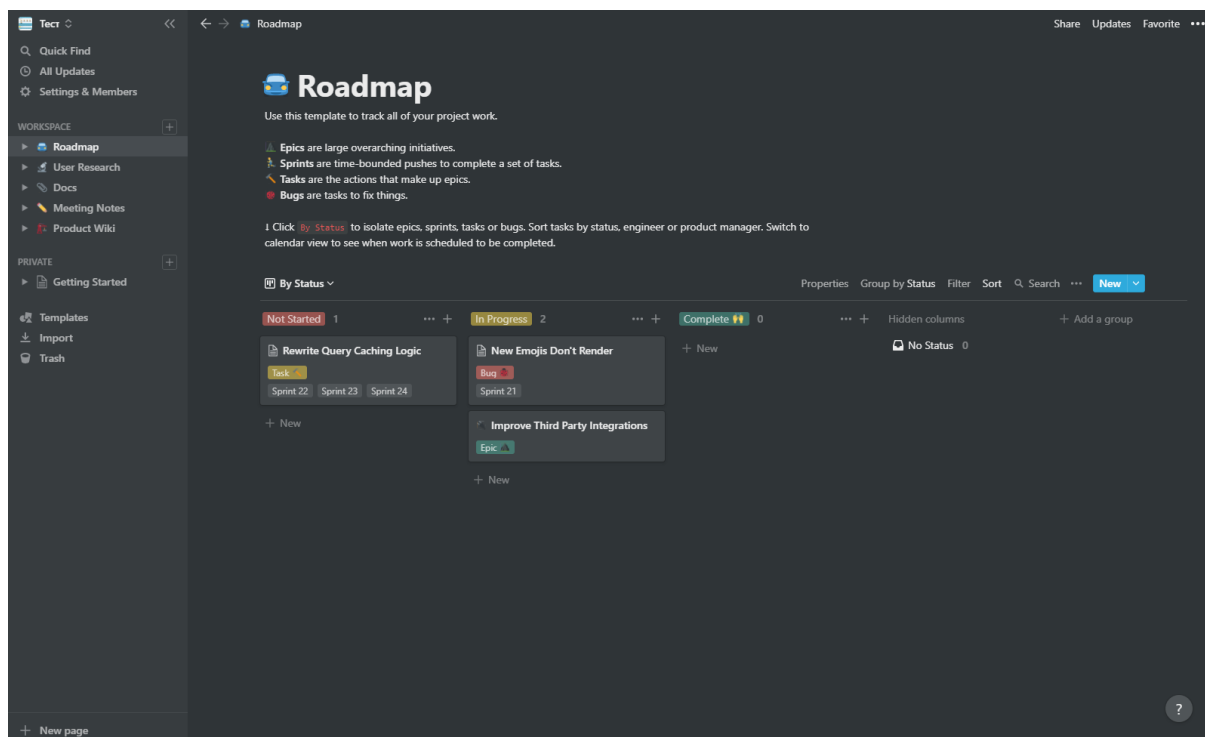


Рисунок 11 - Notion. Kanban доска

В левой части экрана находятся страницы проекта, такие как документация, заметки после встреч или вики по функционалу. На рисунке 12 представлено описание вики с распределением по категориям проекта.

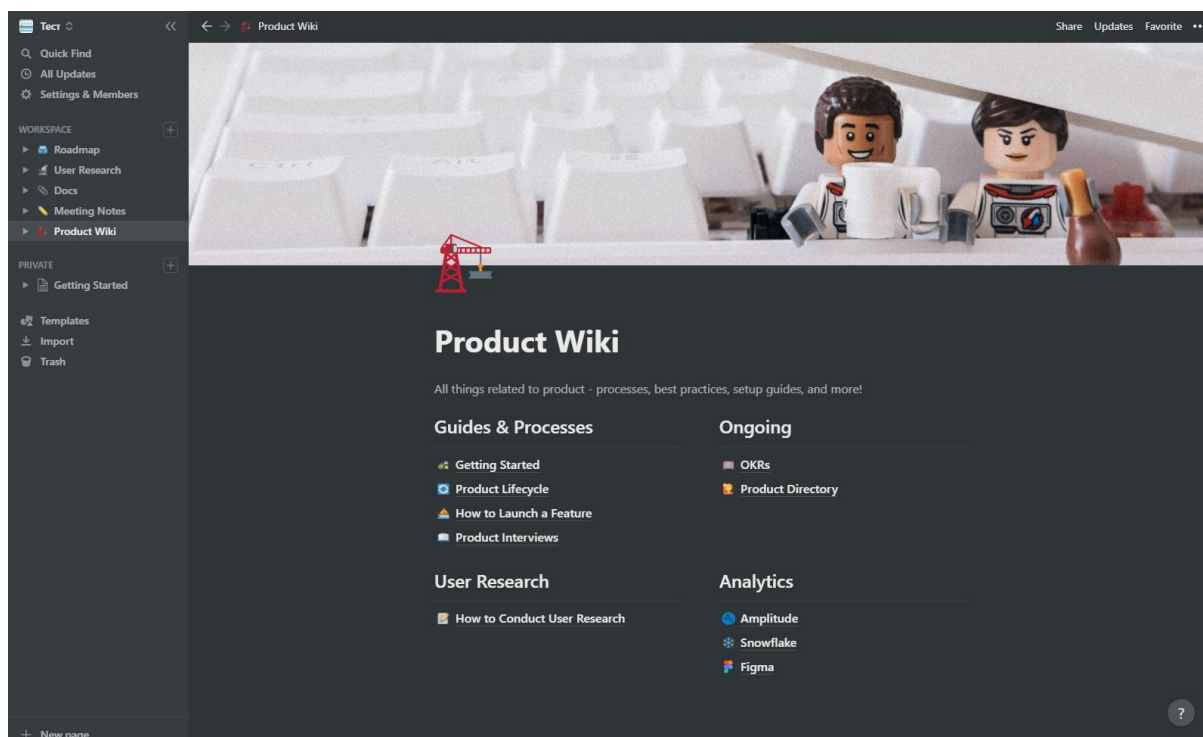


Рисунок 12 - Notion. Wiki.

При нажатии на любой из пунктов в вики открывается отдельная страница с описанием функционала. Notion использует свою разметку для создания документации в отличие от Jira и Assana, где используется Markdown - облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком. На рисунке 13 представлен такой подход, который позволяет пользователю без навыков разметки создавать сложные конструкции за быстрый промежуток времени. Для этого пользователю нужно поставить символ «/» в документе и выбрать нужный элемент. Элементами в документах являются: заголовки, списки, ссылки на страницы внутри рабочего пространства, формулы, таблицы, календари, видео, ссылки на git-

репозитории, файл из систем для реализации прототипов Figma, Abstract, Invasion и многое другое.

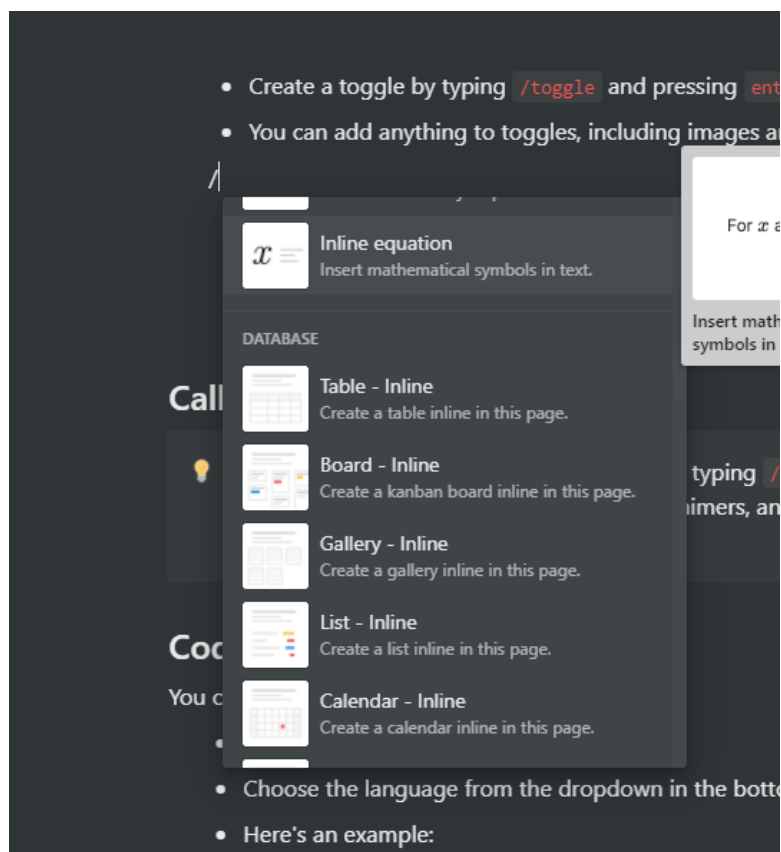


Рисунок 13 - Notion. Вставка элемента на странице

Преимущества использования рабочего пространства Notion:

- 1) интуитивно понятный интерфейс. Все действия по добавлению страниц используют знаки «+» в интерфейсе, или «/» на страницах, далее система подсказывает возможные действия;
- 2) своя разметка, при создании задач, не нужно проверять корректность написания блоков для вставки списка, ссылок, кода - интерфейс сразу отображает итоговый результат ввода;
- 3) вики и доски в одном месте - возможность ссылок на документацию, заметки после встреч, код и медиа-файлы внутри задачи;

- 4) возможность ведения собственных страниц внутри рабочего пространства на вкладке private;
- 5) отсутствуют ограничения на количество страниц в пространстве. Notion подойдет как для небольшого, так и для крупного описания проекта;
- 6) возможность оффлайн режима и обмена документами посредством выгрузки пространства в файл. Можно использовать оффлайн пространство и передавать пространство через защищенные каналы, например в приватный github репозиторий;
- 7) возможность персонализации страниц для визуального разделения страниц, задач, людей.

Недостатки использования Notion:

- 1) ограничение на команду в бесплатном режиме в 5 человек;
- 2) каждая команда в крупном проекте создает свое пространство, отсутствует возможность отследить итоговую картину по всему проекту.

Выводы к главе 2

На основании проведенного исследования инструментов можно сделать следующие выводы. Asana и Jira подходят для крупных проектов с несколькими командами разработки, используют разметку Markdown необходимую для изучения, для взаимодействий с задачами системы.

Для небольшой команды разработки в стартапе, оптимальным решением будет использование Notion, ввиду возможности конфигурации рабочего пространства под свои нужды. Есть возможность создать страницу с договоренностями с бизнесом, страницу с идеями и другими элементами стартапа. Весь проект с документацией хранится в одном месте и не нуждается в дополнительных инструментах, в отличие от мощного промышленного решения Jira.

3 Формирование проекта «Мониторинг IT конференций»

На текущий момент отсутствуют сервисы, предоставляющие материалы прошедших конференций в одном месте. Пользователям приходится необходимую информацию (запись лекции, материалы) искать на сайтах конференций и мероприятий. Организаторы конференций не всегда размещают видео-лекции на своих сайтах. Пользователям приходится искать видео в сети Интернет. Это легко сделать, если пользователь знает название доклада и спикера, но появляются проблемы, когда интерес пользователя предоставляет определенное направление, например, «всё про Python», с прошедшей конференции. В приложении А представлена модель «AS IS» - модель отображает как сейчас пользователи ищут необходимые материалы, в нотации BPMN 2.0 поиска видео с IT-конференций «DUMP».

Сервис «Мониторинг IT конференций» предназначен для сотрудников и работодателей компаний для поиска материалов конференций, а также для пользователей в сфере IT, которые следят за выступлением и докладам коллег. Командой разработки реализуется сервис, в котором будут ссылки на видео и ссылки на локальное расположение файлом с распределением по конференциям и системой тегов для быстрого поиска необходимого контента. В приложении Б представлена модель «AS BE», которая отражает процесс, описанный в приложении А, при помощи реализуемого сервиса «Мониторинг IT конференций».

Для реализации проекта необходимо определить процесс жизненного цикла информационной системы. В команде задействованы 3 человека, роли для распределения нагрузки между членами команды определены следующим образом: 1 - аналитика и дизайн, 2 - разработка backend и devops, 3 - разработка frontend и devops. Итоговая модель разработки представляет 3 статуса:

- 1) «Аналитика и Дизайн» - техническое описание логики процессов, построение ER-диаграмм, реализацию прототипов сервиса;

- 2) «Разработка» - включает в себя разработку пользовательского интерфейса на основе макетов - Frontend, реализацию кодовой базы на основе описания технической части - Backend;
- 3) «Сделано» - функционал добавлен в проект на gitlab.

На основании главы 1 методологией управления проектом является kanban, на основании главы 2 средством управления проектом выбран Notion. Реализована рабочая доска проекта представленная на рисунке 14.

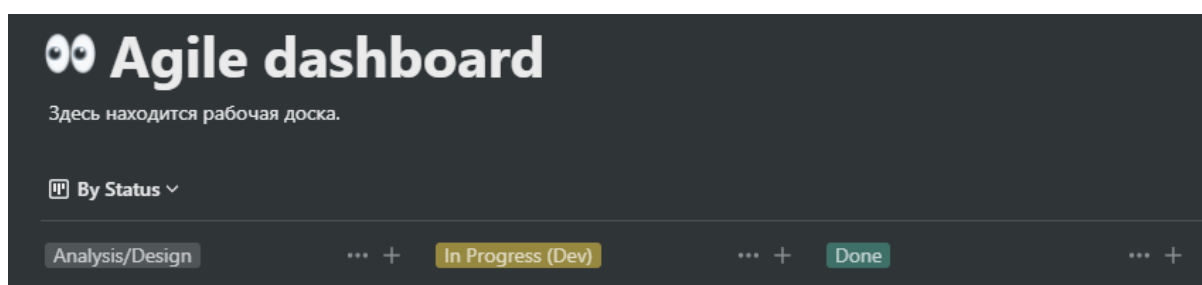


Рисунок 14 - Статусы рабочей доски проекта

Для реализации проекта необходимо использовать облачные сервера с кластером kubernetes. В связи с необходимостью привлечения денежных средств, установлен минимальный бюджет проекта в размере 10 000 Р.

Для реализации итеративной разработки был установлен срок выполнения задач в рамках одного sprint длительностью один календарный месяц.

В ходе управления проектом, были сформированы требования к проекту представленные ранее. В таблице 4 представлена сводная таблица проекта, отражающая ключевые параметры: бюджет, итерации, участников и роли.

Таблица 4 - сводная таблица проекта

Бюджет проекта	10 000 Р
Периодичность итераций	Каждое пятое число месяца

Продолжение таблицы 4

Участники и роли	Акулич Сергей - аналитика, дизайн Сухарев Никита - backend, devops Савичев Игорь - frontend
Статусы задач в проекте	«Аналитика и Дизайн» «Разработка» «Сделано»
Срок реализации	365 дней

Выводы к главе 3

В рамках реализации проекта построены модели в нотации BPMN, представленные в приложении А, Б, с помощью которых выявлен способ решения проблемы поиска видео с IT-конференций.

Реализована рабочая доска проекта, выделены ключевые роли для каждого из участников с распределением обязанностей. Выбран срок реализации, периодичность итераций и бюджет проекта.

4 Прототипы сервиса

Для реализации минимального жизнеспособного продукта (MVP) необходимо выполнен следующий план работ:

- 1) создание style guide UI части приложения;
- 2) авторизация пользователя;
- 3) хранение, отображение и категоризация видео-контента;
- 4) отметка избранного видео пользователем.

Для реализации макетов используется онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени - Figma.

Для создания гайдлайна, который будет соответствовать требованиям поставленной задачи, необходимо обратиться к исследованиям Kissmetrics[25]:

92.6% людей заявляют, что визуальный аспект больше всего влияет на решение о покупке, сильнее, чем вкус, запах и т.д.

Два из трех потребителей не купят крупную бытовую технику, если ее нет в нужном цвете.

В другом исследовании под названием Impact of Color in Marketing исследователи обнаружили, что 90% суждений о других продуктах основаны только на цвете. Респондентов попросили выбрать цвет, ассоциирующийся с конкретными словами.

Доверие: большинство выбрали синий (34%), реже белый (21%) и зеленый (11%);

Безопасность: синий (28%), черный (16%) и зеленый (12%);

Скорость: абсолютным фаворитом оказался красный (76%);

Дешевизна: оранжевый (26%), желтый (22%) и коричневый (13%);

Высокое качество: черный стал абсолютным победителем (43%), затем шел синий (20%);

Высокие технологии: тут мнения разделились между черным (26%) синим и серым (оба 23%);

Надежность: синий (43%), черный (24%);

Храбрость: фиолетовый (29%), красный (28%), синий (22%);

Ужас/страх: красный (41%), черный (38%);

Смех: оранжевый был фаворитом (28%), желтый стал следующим (26%), и фиолетовый после него (17%).

Когда речь идет о надежности и доверии, синий станет лучшим выбором. Соответственно, приложения, которым люди доверяют личную информацию, например, Facebook, Twitter, LinkedIn, Skype, Вконтакте, Госуслуги – раскрашены в синий цвет не случайно.

В соответствии с описанным исследованием было принято решение использовать синие цвета в качестве основной цветовой схемы приложения. Цветовая схема представлена на рисунке 14.

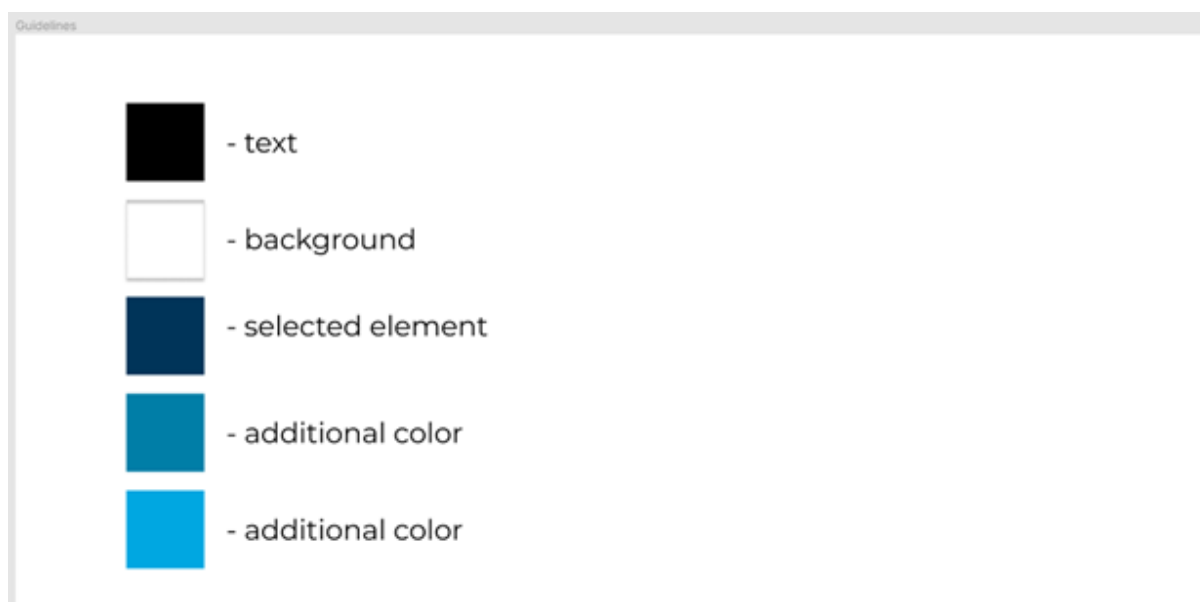
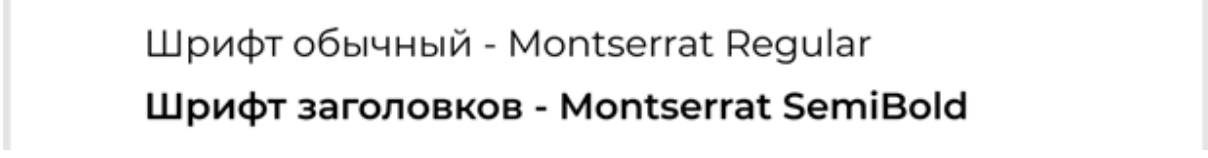


Рисунок 14 – Style guide. Цветовая схема приложения

В задаче реализации Style guide продуманы используемые шрифты, представленные на рисунке 15. В качестве основных шрифтов были выделены

деловые шрифты Montserrat Regular для подчеркивания индивидуальности проекта, не является широко распространённым шрифтом и SemiBold версия для выделения заголовков.

The image shows a style guide with two lines of text. The first line is 'Шрифт обычный - Montserrat Regular' in a regular weight. The second line is 'Шрифт заголовков - Montserrat SemiBold' in a bold weight. The text is centered between two vertical grey bars.

Шрифт обычный - Montserrat Regular
Шрифт заголовков - Montserrat SemiBold

Рисунок 15 – Style guide. Основные шрифты приложения

Авторизация. В задачу реализации страниц авторизации необходимо включить стандартную форму авторизации, с привычными для пользователя полями, такими как логин и пароль. В случае если пользователю предоставят для заполнения личные данные, такие как Имя и Фамилия, он может неоднозначно понять необходимые для ввода данные и покинуть сервис, не дойдя до страниц с видео.

Таким образом были реализованы страницы входа по логину пользователя и страницы регистрации, представленные на рисунках 16-17 соответственно, с использованием разработанного Style guide приложения.

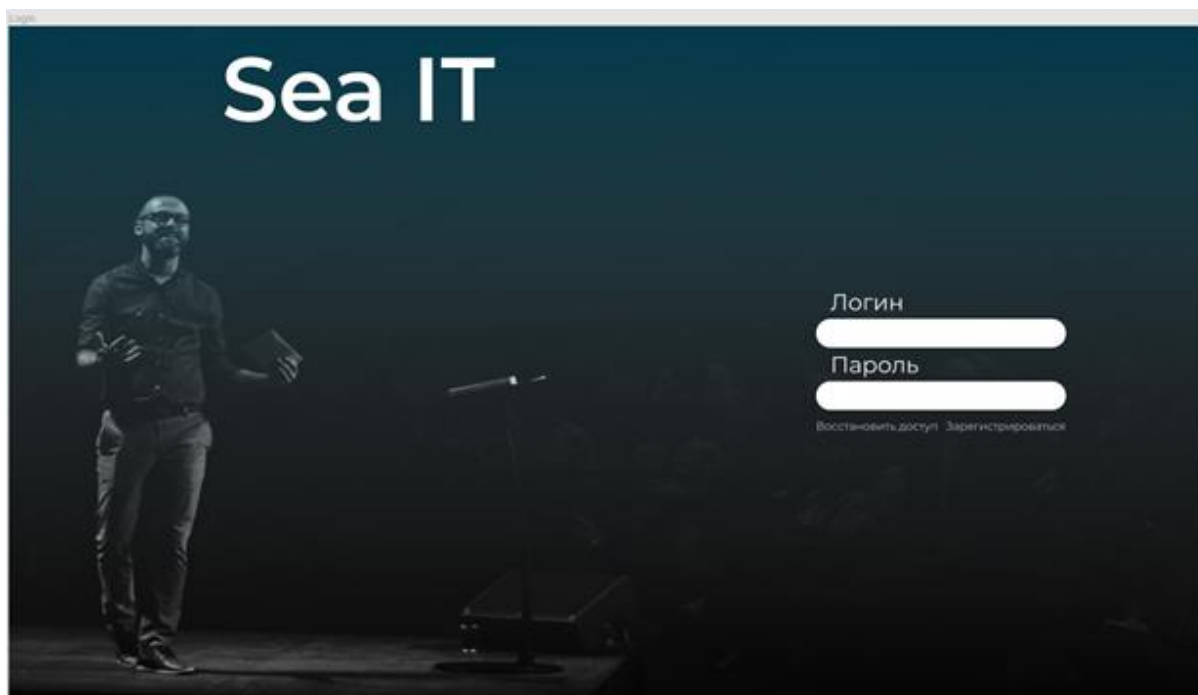


Рисунок 16 – Авторизация в приложении

Для того, чтобы пользователь обратил внимание на ошибки при вводе данных реализованы `snackbar`, отражающие текущее состояние действий клиента. Так на рисунке 4, пользователь заметит, что совершил ошибку при вводе данных, с указанием на поля, в которых совершена ошибка. В случае с логином и паролем, указывается оба поля, для создания дополнительной UI безопасности внутри приложения.

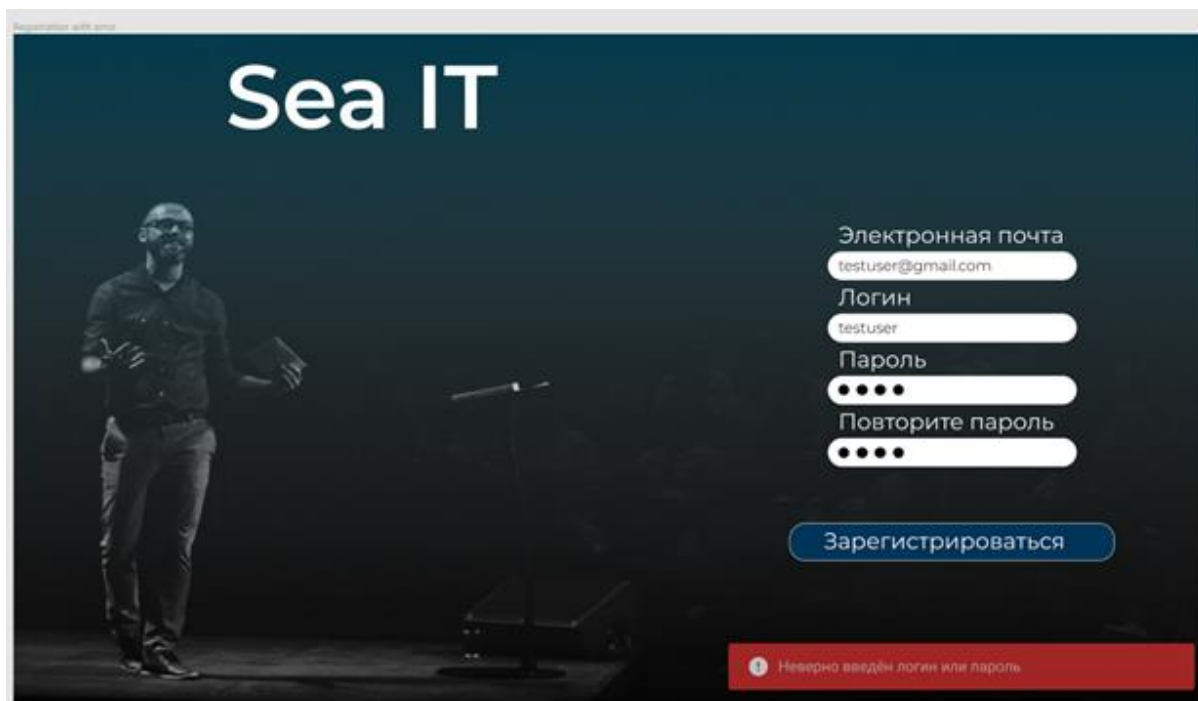


Рисунок 17 – Регистрация в приложении с snackbar-ошибкой.

Внутри всего приложения snackbar являются основным элементом взаимодействия с пользователем. Они используются в случае возникновения ошибок, а также предполагается использование в случае позитивных сообщений, таких как появление нового видео, по интересующей клиента категории, успешных событиях смены почты, логина и других. На рисунке 18 представлены snackbar, рекомендованные к использованию material.io[26].

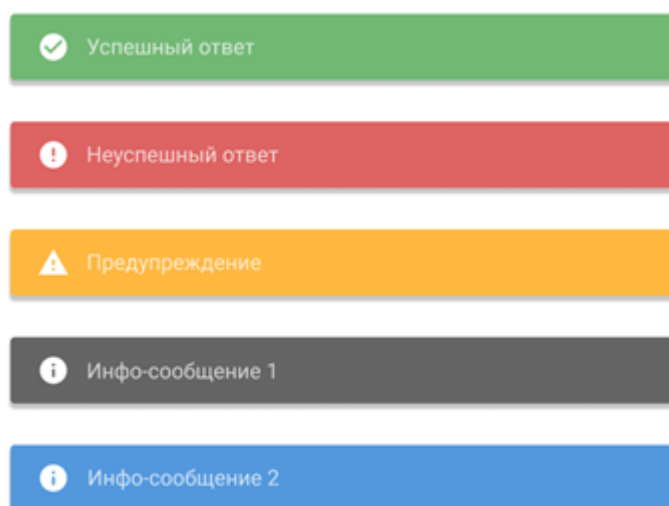


Рисунок 18 – Snackbar в приложении.

Стандартные иконки

При разработке приложений существуют вероятность, когда нет возможности получить фотографию спикера, логотип технологии или конференции в открытом доступе, для решения проблемы были реализованы иконки, которые будут заменять пустое пространство, чтобы сервис казался пользователю полноценным и наполненным. На рисунке 19 представлены стандартные иконки в приложении, слева направо: иконка для отображения конференций, спикеров, технологий.



Рисунок 19 – Стандартные иконки

Навигация по сервису

После того, как пользователь прошел процедуру регистрации, он попадает на главную страницу сервиса. На главной странице отображены конференции представленные в сервисе. В верхней части экрана присутствует меню навигации для быстрого перемещения между категориями сайта. Выбранная категория подчеркнута. Рисунок 20 отражает главную страницу сервиса, которую пользователь видит при первом входе.

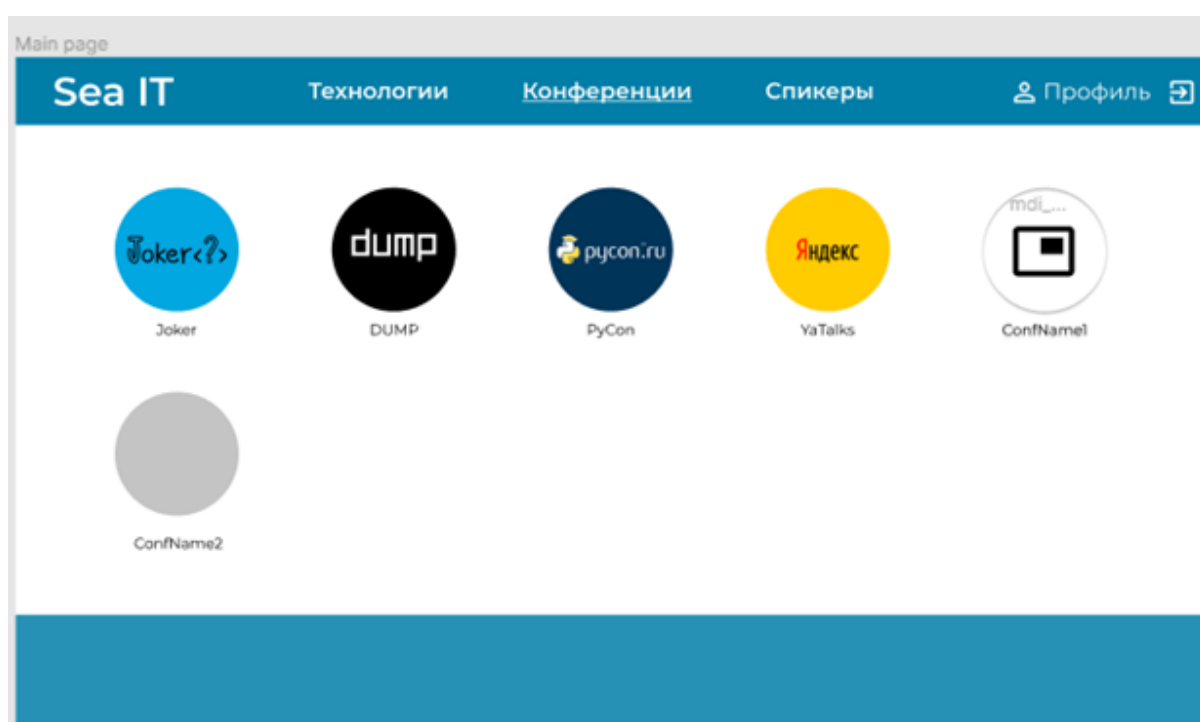


Рисунок 20 – Главная страница сервиса

При выборе рубрики в категории пользователь попадает на страницу с видео, представленную на рисунке 21. Верхняя часть динамически меняется в зависимости от категории и рубрики, в случае с категорией «Конференции» шапка содержит информацию о конференции, например DUMP, в случае, если выбрана категория «Технологии», в шапке название технологий и технологий ответвлений, например Python (Django, Machine Learning, Flask), в случае если выбрана категория «Спикеры», в шапке будет предоставлена информация,

какими технологиями владеет спикер, например Антон Верднев (Python, docker, kubernetes, django).

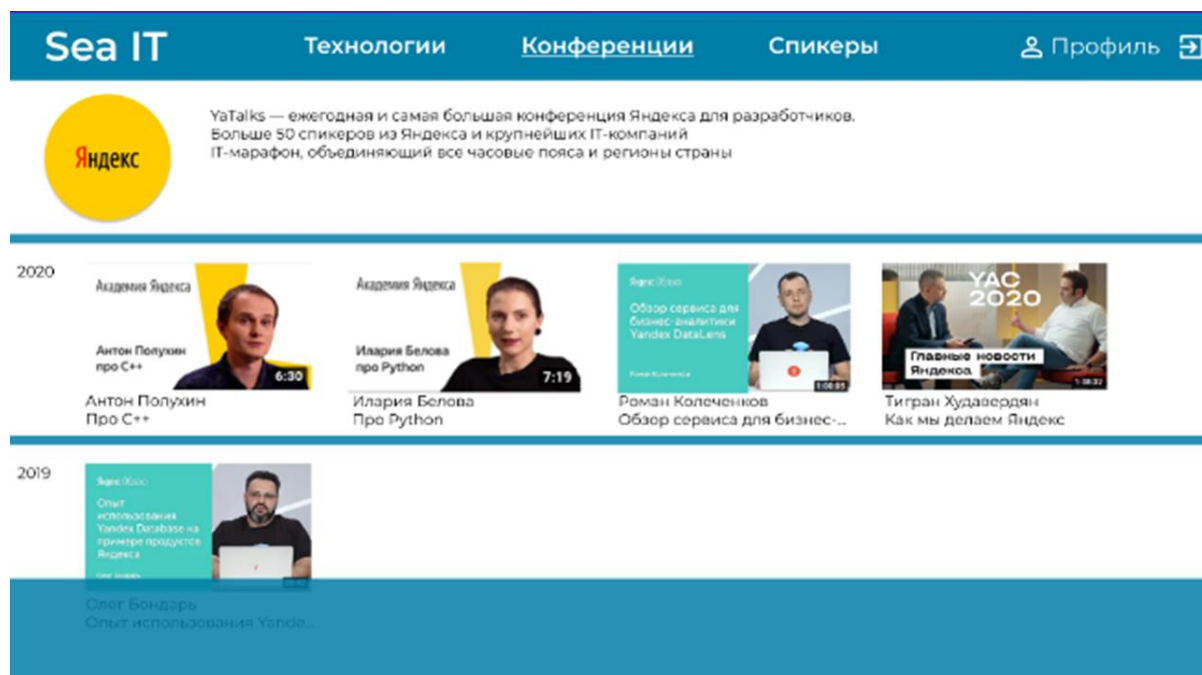


Рисунок 21 – Страница с видео

Для реализации личного кабинета пользователю необходимо отображать следующие компоненты:

- 1) компонент «изменения данных»;
- 2) блок «избранное с видео», после просмотра видео у пользователя есть возможность добавить видео в избранное, если пользователь совершил данное действие, видео отобразится в личном кабинете в блоке;
- 3) блок «избранное для новых пользователей», отображение блока, когда пользователь еще не отметил видео избранным.

После реализации каждого из компонентов были получены макеты, представленные на рисунках 22 и 23. На рисунке 22 в правой верхней части реализован компонент «изменения данных» пользователя, который

включается в себя поле для ввода данного на изменение и кнопку для совершения действия, например «Изменить почту».

The screenshot shows the 'Sea IT' user profile page. The header includes the 'Sea IT' logo and navigation links: 'Технологии', 'Конференции', 'Спикеры', and 'Профиль'. The 'Настройки' (Settings) section displays the user's login as 'new_user' and email as 'new_user@gmail.com', with input fields and buttons to change them. The 'Избранное' (Selected) section contains three video thumbnails: 'Вакс Юло' by Роман Колечников, 'Академия Яндекса' by Илария Белова, and 'YAC 2020' by Тигран Худавердян. A green banner at the bottom states: 'Для подтверждения действия перейдите по ссылке в почте'.

Рисунок 22 - Личный кабинет пользователя с заполненным блоком “Избранное”

На рисунке 23 представлен блок в профиле, когда пользователь не добавил видео в избранное.

The screenshot shows the 'Sea IT' user profile page with the 'Избранное' (Selected) section empty. The settings and navigation elements are identical to Figure 22. Below the 'Избранное' section, a message reads: 'Добавляйте видео в избранное, чтобы посмотреть оно отобразилось в прифле'. A large blue rectangular area is present at the bottom of the page.

Рисунок 23 - Личный кабинет пользователя с пустым блоком “Избранное”

В интерфейсе были реализованы snackbar, отвечающие за информирование пользователя о совершенном действии. На рисунке 22 представлено использование snackbar при успешном совершении операции пользователем – смены почты и пароля.

Для подчеркивания изменяемой информации используется графический элемент «галочка», который помогает пользователю сориентироваться, какие данные профиля были изменены.

Блок избранное масштабируется в зависимости от количество добавленного и отмеченного пользователем видео. Блок превращается в стандартную страницу, где видео можно листать до конца списка.

В блоке, где отсутствует избранное видео, пользователю отображается подсказка об изменяемости элемента интерфейса, в зависимости от его действий.

Выводы к главе 4

Реализована дизайн система будущего проекта. На основании иностранных исследований определена цветовая гамма, удовлетворяющая потребности сервиса в увеличении доверия. Представлены прототипы системы для реализации верстки frontend части приложения.

5 Проектирование системы

Для реализации проекта необходимо рассмотреть пользовательские действия внутри приложения, реализовать пользовательские истории к каждому действию, описать структуру в базе данных посредством ER-диаграмм.

Для того, чтобы получить доступ к видео материалам пользователю необходимо зарегистрироваться в системе. Пользовательская история:

- 1) пользователь нажимает кнопку зарегистрироваться;
- 2) заполняет данные на форме регистрации: логин, почту и пароль;
- 3) пользователю на указанную при регистрации почту отправляется ссылка подтверждение учетной записи, при переходе по которой активируется учетная запись.

В реализации используется шаблон таблиц регистрации и авторизации через фреймворк Django [27] с модернизацией посредством JSON Web Token [28]. На рисунке 24 представлены связи в таблице регистрации и авторизации, включая отдельные таблицы для администрирования системы. В стандартную таблицу `authorizer_user` добавлены параметры:

- 1) `login` - логин пользователя, указанный при регистрации
- 2) `email` - электронная почта пользователя, указанная при регистрации
- 3) `password` - пароль, указанный пользователем при регистрации
- 4) `confirmation_hash` - hash-значение отправленное в ссылке на электронную почту при регистрации, необходимое для подтверждение электронной почты
- 5) `confirmation_at` - дата отправки ссылки-подтверждения электронной почты, для реализации истечения времени действия ссылки

- 6) status - статус пользователя имеет 3 типа: inactive (неактивированная учётная запись), active (активная учётная запись), deactivated (удаленная/деактивированная учётная запись)

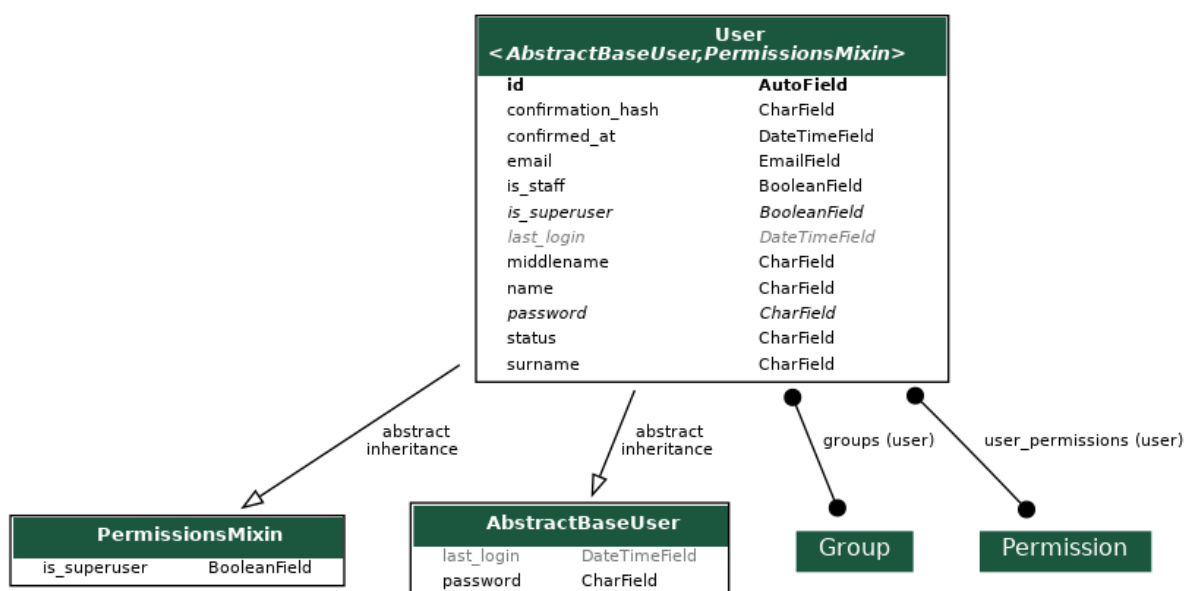


Рисунок 24 - ER-диаграмма таблиц регистрации и аутентификации

В рамках задачи регистрации необходимо реализовать HTTP POST метод «регистрация» в формате JSON - /registration. В таблице 5 описаны входные параметры, значения параметров и пользовательский пример.

Таблица 5 - описание входных данных /registration

Входной параметр	Значение параметра	Пример
login	Логин	testlogin
email	Электронная почта	testlogin@test.com
password	Пароль	R4e3w2q1

В случае неуспешного ответа вернется ошибка с указанием на поле, пример ответа представлен на рисунке 25.

```
{
  "email": [
    "Enter a valid email address."
  ],
  "password": [
    "Ensure this field has at least 8 characters."
  ]
}
```

Рисунок 25 - некорректный адрес электронной почты и пароль

В случае успешного запроса:

- 1) параметры записываются в таблицу `authorizer_user`;
- 2) параметр `status` устанавливается в значение `inactive`;
- 3) отправляется письмо на указанную почту со ссылкой на активацию (ссылка выглядит как `{domain}/confirm?h`, где `h` - значение из таблицы `authorizer_user.confirmation_hash`;
- 4) в ответе на запрос вернётся параметр `"confirmed"` со значение `true` или `false` в зависимости от успешности совершения операции. Примеры ответов представлены на рисунке 26.

```
#Пример запроса с неправильным хешем - код ответа HTTP 400
/confirm?h=invalid_hash_from_message
{
  "confirmed": false
}
#Пример положительного ответа: HTTP 200
{
  "confirmed": true
}
```

Рисунок 26 - ответы на запрос `/confirm`

Для реализации сервиса необходимо реализовать таблицы для хранения информации о видео, спикерах, конференциях и технологиях. В таблице 6 представлено описание параметров в базе данных, таблицы конференции.

Таблица 6 - описание таблицы конференций «confs»

Параметр	Значение параметра	Пример
id	уникальный идентификатор	1
name	наименование конференции	DUMP
description	описание конференции	Ежегодная конференция, которая проходит в Екатеринбурге. Проходит по всем направлениям от project managment до devops
icon	логотип конференции	-
icon_hash	хэшированное значение иконки, для проверки на фронте обновлена ли иконка (md5("icon"))	63e1549b7609f7dd1aeb792a7227fb49

В таблице 7 представлено описание параметров в базе данных, таблицы технологии.

Таблица 7 - описание таблицы технологий «technologies»

Параметр	Значение параметра	Пример
id	уникальный идентификатор	1
name	наименование технологии	Docker

Продолжение таблицы 7

parent_name	название родительской технологии	«name = Django → parent_name = Python» «name = Python → parent_name = null»
icon	ЛОГОТИП технологии	-
icon_hash	хэшированное значение иконки, для проверки на фронте обновлена ли иконка (md5("icon"))	63e1549b7609f7dd1aeb792a7227fb49

В таблице 8 представлено описание параметров в базе данных, таблицы спикеры.

Таблица 8 - описание таблицы технологий «speakers»

Параметр	Значение параметра	Пример
id	уникальный идентификатор	1
name	фамилия имя или фамилия имя отчество спикера	Алексей Кулаков
tech_id	область знаний, связь с таблицей technologies	[1, 4, 10]
icon	фотография спикера	null

Продолжение таблицы 8

icon_hash	хэшированное значение иконки, для проверки на фронте обновлена ли иконка (md5("icon"))	null
-----------	--	------

В таблице 9 представлено описание параметров в базе данных, таблицы видео со связями, по каждой из представленных выше сущностей.

Таблица 9 - описание таблицы хранения видео «videos»

Параметр	Значение параметра	Пример
id	уникальный идентификатор	1
link	ссылка на видео	https://www.youtube.com/watch?v=br3mtBZ6HsY
name	наименование видео	Продуктовая разработка и маркетинг на больших поисковых данных
tech_id	область знаний, связь с таблицей technologies	[1, 4, 10]
conf_id	видео с конференции, связь с таблицей confs	14

Продолжение таблицы 9

speaker_id	спикер, связь с таблицей speakers	12
year	год записи видео	2021

На рисунке 27 представлена ER-диаграмма отражающая связи между таблицами. Диаграмма реализована в PlantUML. Листинг кода диаграммы представлен в приложении В

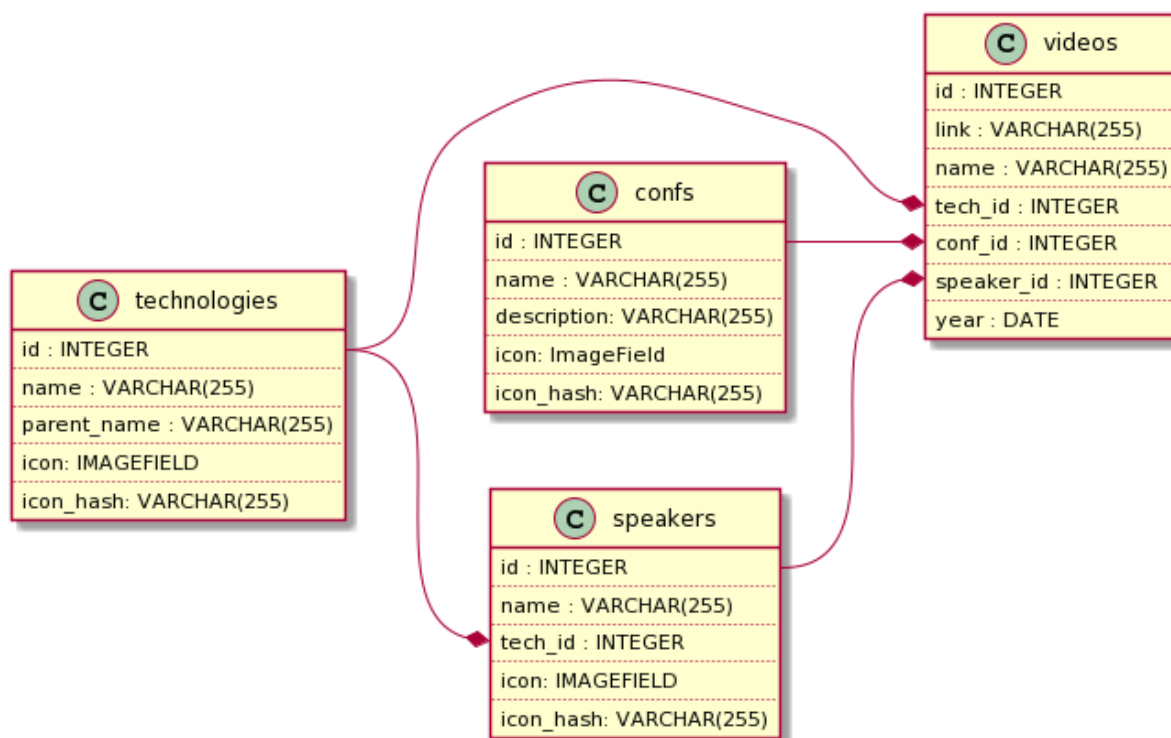


Рисунок 27 - ER-диаграмма связей таблиц видео и категории

Для взаимодействия с интерфейсной частью приложение необходимо реализовать методы для получения сущности «Категории», «Спикеры», «Технологии», а также метод для получения ссылки на видео, либо видео в зависимости от способа хранения информации в базе.

Необходимо реализовать метод {url}/get?type, в качестве входного параметра могут использоваться:

- 1) confs;
- 2) technologies;
- 3) speakers;

Примеры: www.seait.ru/get?type=confs, www.seait.ru/get?type=speakers
www.seait.ru/get?type=technologies

В таблице 10 представлены значения параметров ответа с описанием в зависимости от входного параметра.

Таблица 10 - входные параметры метода get

Параметр	Описание	Тип данных
входной параметр confs		
id	уникальный идентификатор	integer
name	наименование конференции	string
description	короткое описание конференции	string
icon	ссылка на логотип конференции	string
входной параметр speakers		
id	уникальный идентификатор	integer
name	фамилия, имя, отчество (если есть) спикера	string
tech_id	связанные со спикером технологии	integer
icon	ссылка на фото спикера	string

Продолжение таблицы 10

входной параметр technologies		
id	уникальный идентификатор	integer
name	название технологии	string
parent_name	название основной (родительской) технологии	string
icon	ссылка на логотип технологии	string

Для получения видео необходимо реализовать метод /videos, где входными параметрами являются:

- 1) type - тип, может принимать 3 значения: technology, speaker, conf;
- 2) id - уникальный идентификатор технологии, спикера или конференции.

В таблице 11 представлены параметры в ответе на запрос /videos в зависимости от параметра type

Таблица 11 - описание ответов на запрос /videos

Параметр	Описание	Тип данных
Входной параметр conf. Поиск в базе данных по conf_id		
link	ссылка на видео	integer
name	наименование видео	string
description	короткое описание конференции	string
icon	ссылка на логотип конференции	string
year	год появления видео	string

Продолжение таблицы 11

tech_id	уникальный идентификатор технологии, о которой рассказывают в видео	string
conf_id	уникальный идентификатор конференции, на которой был представлен материал	string
speaker_id	уникальный идентификатор спикера	string
Входной параметр speaker. Поиск в базе данных по speaker_id		
link	ссылка на видео	integer
name	фамилия, имя, отчество (если есть) спикера	string
technologies	технологии, которыми владеет спикер	array
icon	ссылка на логотип конференции	string
year	год появления видео	string
tech_id	уникальный идентификатор технологии, о которой рассказывают в видео	string
conf_id	уникальный идентификатор конференции, на которой был представлен материал	string
speaker_id	уникальный идентификатор спикера	string

Продолжение таблицы 11

Входной параметр technology. Поиск в базе данных по tech_id		
link	ссылка на видео	integer
name	наименование технологии	string
parent_name	название основной (родительской) технологии	array
icon	ссылка на логотип конференции	string
year	год появления видео	string
tech_id	уникальный идентификатор технологии, о которой рассказывают в видео	string
conf_id	уникальный идентификатор конференции, на которой был представлен материал	string
speaker_id	уникальный идентификатор спикера	string

На рисунке 28 представлен пример ответа на запрос videos. В первом случае в базе данных сохранена ссылка на видео во внешнем источнике, во втором случае видео хранится на нашем сервере.

```

#Пример ответа
{
  "name": "Python"
  0:
  {
    "link": "www.youtube.com/ASjkaAfa12Fa2Fk"
    "year": "2019"
    "conf_id": 10
    "tech_id": 2
    "speaker_id": 93
  }
  1:
  {
    "link": "ceph-hash/videos/name.mp4?AWSAccessKeyId="
    "year": "2019"
    "conf_id": 13
    "tech_id": 2
    "speaker_id": 103
  }
}

```

Рисунок 28 - примеры ответов на запрос /videos

Выводы к главе 5

Командой разработки был реализован проект «Мониторинг IT-конференций», который дает возможность эффективного нахождение информации с конференций в Интернете.

Реализован функционал сервиса:

- 1) доступ к видео в категории “Технологии”, внутри которой, пользователь выбирает основную категорию - язык программирования, например: Python, а также подкатегорию - технологию, например: Django. При работе с необычными, новаторскими технологиями, например, DevOps, IT-специалист может быстрее получить информацию по смежной системе с обсуждением похожей проблемы.
- 2) доступ к видео в категории “Конференции”. Сбор полной записи всех выступлений, материалов в одном месте, так как обычно выступления могут идти одновременно, пользователь должен имеет возможность посмотреть все выступления с одного мероприятия.
- 3) доступ к видео в категории “Спикеры”, с выбором категории по фамилии, имени докладчика на конференции. Пользователь выбирает спикера и смотрит все доклады конкретного человека. Данный функционал может помочь с рекрутингом в IT - найти кандидатов для работы, а также повышает личный бренд каждого спикера.

6 Экономическое обоснование

При реализации приложения необходимо учитывать затраты на используемые ресурсы. Все расчеты произведены на 2021 год. Для размещения приложения приобретаются облачные виртуальные машины на серверах компании Yandex. [29] Для реализации проекта необходимо 2 виртуальные машины. Одна виртуальная машина с названием «Gitlab Runner» необходима для сборки проекта. Характеристики виртуальной машины представлены в таблице 12.

Таблица 12 – Характеристики виртуальной машины Gitlab Runner

Операционная система	Ubuntu 20.04
Объем памяти жесткого диска	5 ГБ
Процессор	Intel Cascade Lake 50% процессорного времени
Количество ядер	2
Объем оперативной памяти	2 ГБ
Публичный ip-адрес	+

Также необходима виртуальная машина «App» для развертывания проекта, хранения видео-контента, к которой будут подключаться пользователи. К виртуальной машине необходимо купить доменное имя подходящее под стилистику, реализованной дизайн системы. Характеристики машины «App» представлены в таблице 13

Таблица 13 – Характеристики виртуальной машины App

Операционная система	Ubuntu 20.04
Объем памяти жесткого диска	200 ГБ

Продолжение таблицы 13

Процессор	Intel Cascade Lake 50% процессорного времени
Количество ядер	4
Объем оперативной памяти	8 ГБ
Публичный ip-адрес	Дополнительный оплачиваемый параметр
Доменное имя	seait.ru

На виртуальной машине 1 «Runner» реализуется сборка проекта, стоимость виртуальной машины составляет 1050 рублей в месяц. На виртуальной машине 2 «App/kubernetes» происходит развертывание приложения, стоимость виртуальной машины составляет 3000 рублей. Также для проекта необходимо приобрести доменное имя стоимостью 300 рублей в месяц. В ситуации, когда другие расходы на производство отсутствуют, расходы на реализацию проекта составляют 4050 рублей в месяц без покупки домена на время разработки и 4350 рублей при запуске проекта.

В момент разработки приложения было принято решение сократить расходы на машину с развернутым приложением, уменьшив размер жесткого диска до 50 ГБ, а также отключать виртуальные машины на время отсутствия работы с ними. По итогу во время разработки, траты на приложения не превышали 2000 рублей в месяц.

Разработка инфраструктуры, непосредственной работы с виртуальными машинами длилась на протяжении трех месяцев и на текущий момент траты составляют 5274,58 рублей. Инфраструктура проекта представлена на рисунке 29.

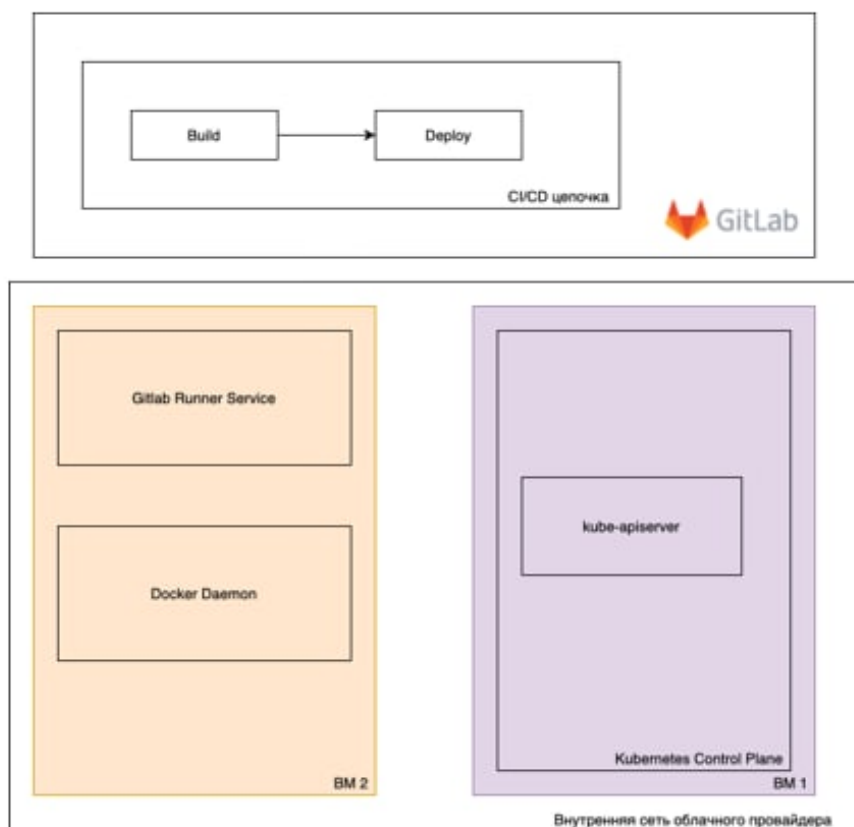


Рисунок 29 - Инфраструктура проекта

Для восполнения расходов выбраны 3 модели реализации экономического потенциала проекта. Каждая модель не учитывает государственные пошлины за регистрацию юридического лица и рекламу. Расчет происходит из упрощенной системы налогообложения, которая составляет 6%, с возможностью регистрации одного из членов команды в качестве индивидуального предпринимателя.

Первая модель включает в себя разрабатываемый сервис с системой подписок, где стоимость одной подписки будет составлять 179 рублей и давать полный доступ к эксклюзивным материалам с прошедших IT-конференций. После вычета налогов остается 155,73 рубля дохода. Также необходимо учитывать процент компаниям за предоставленное видео, а именно 50 рублей за каждую покупку клиента. Чистый доход от продажи подписок составит 105,73 рублей. Таким образом, приложение, которое обходится в 4350 рублей

в месяц, должно иметь 42 уникальных пользователя оформивших подписку на платные услуги сервиса, без учета рекламы и госпошлин за регистрацию юридического лица.

Вторая модель включает в себя продажу сервиса компании, которая занимается организацией конференций. При тратах на реализацию проекта в 5274,58 рублей и возможных доработках в части функционала сервиса, предположим увеличение расходов до 8000 рублей, рассчитаем стоимость сервиса. При совершении сделки продажи интеллектуальной собственности, команда получит прибыль с каждого рубля, при сумме сделки в 8511 рублей.

Третья модель подразумевает изменение концепции сервиса, изменение текущей структуры методов с сохранение инфраструктуры виртуальных машин. Сервис «Мониторинг IT-конференций» с реализованных функционалов хранения ссылок, видео на сервере и предоставление в режиме онлайн может быть переиспользован в обучающих проектах, в развлекательных проектах и других, где необходимо предоставлять информацию посредством видеопотока.

Выводы к главе 6

Командой разработан сервис, который является коммерчески выгодным на рынке, имеет несколько моделей экономической прибыли и решает потребность видеохостинга на сторонних площадках в случае приобретения данного решения.

ЗАКЛЮЧЕНИЕ

В соответствии с целью исследования, было проведено теоретическое обоснование модели управления проектом «Мониторинг IT-конференций» на основании анализа существующих методологий. Был проведен анализ источников по теме исследования: научных статей, учебных пособий по теории управления проектами, материалов в сети Интернет, где содержится большое количество значимой информации о практическом опыте использования различных информационных систем управления проектами, что позволило теоретически обосновать необходимость и возможность использования существующих средств и инструментов для построения моделей проекта.

Для достижения поставленной цели были изучены современные методы, инструменты и информационные системы управления проектами, определены их преимущества и недостатки; были проанализированы методологии управления проектами, с выявлением недостатков и преимуществ для реализации проекта в компании до 5 человек, проанализированы популярные средства управления проектами, которые используются на промышленном производстве и в маленьких компаниях, выбран лучший инструмент для конкретной команды, с описанием результатов использования.

Теоретическое обоснование и анализ процессов поиска видео в Интернете позволили реализовать модель в нотации BPMN «AS IS» существующих решений организации видео сервисов, предоставляющих медиа информацию с IT-конференций. На основании проведенного анализа выделена модель в нотации BPMN «To Be», реализованного сервиса, с определением решения проблемы поиска некачественно категоризированной информации, с прошедших конференций, в Интернете.

Для реализации проекта команды была произведена верстка макетов, реализован style guide для проектируемого сервиса. Для каждой части системы

реализовано техническое описание. Реализованы экономические модели коммерциализации проекта, с учетом текущих расходов.

Рекомендации по практическому применению сервиса и дальнейшее развитие заключаются в следующем:

- 1) сервис оплаты, для возможности принятия платежей за использование материала;
- 2) система подписок, внутри которых хранятся эксклюзивные материалы, предоставляемые пользователям за вознаграждение;
- 3) онлайн-трансляция с мероприятий, для возможности подключения во время проведения конференций;
- 4) календарь индивидуальных мероприятий, для предоставления пользователю информации о предстоящих событиях в сообществе, связанных с интересующей потребителя тематикой;
- 5) система уведомлений посредством PUSH или СМС сообщений

Приложение показало себя конкурентоспособным по сравнению с аналогичными сервисами по предоставлению медиа информации. Достоверность и обоснованность результатов обеспечивается выбранной теоретической и методологической основой исследования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

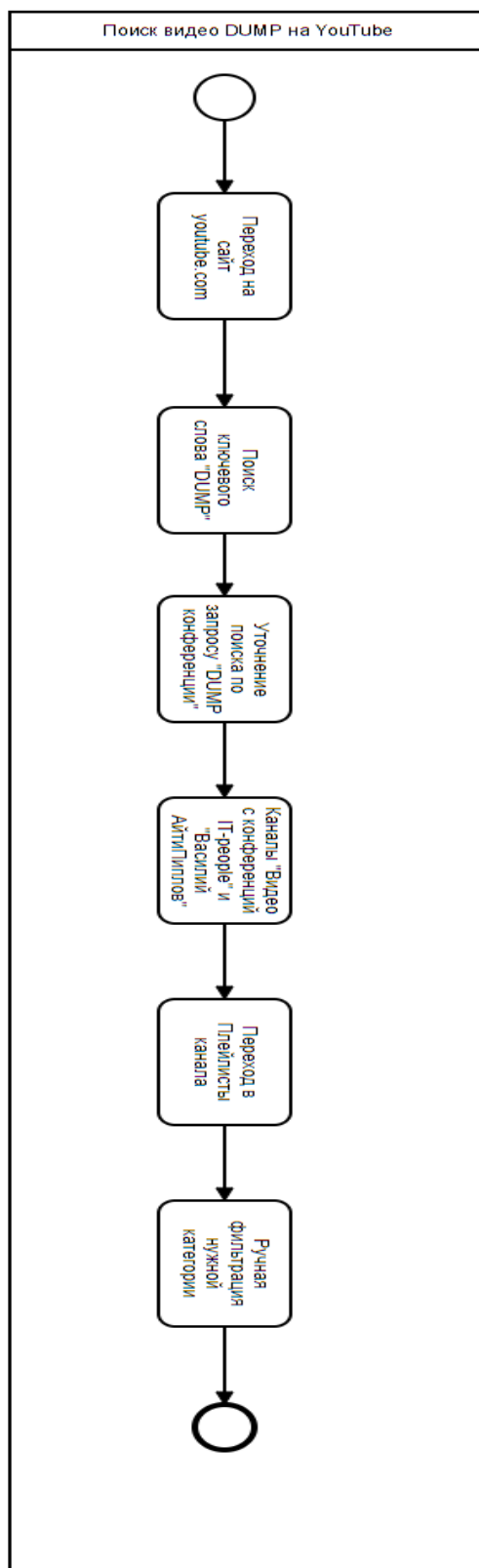
- 1) Сервис онлайн онлайн-видеокурсов [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://www.udemy.com/>], свободный. — (дата обращения 18.02.2021).
- 2) Вся статистика интернета и соцсетей на 2021 год — цифры и тренды в мире и в России [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://www.web-canape.ru/business/vsya-statistika-interneta-i-socsetej-na-2021-god-cifry-i-trendy-v-mire-i-v-rossii/>], свободный. — (дата обращения 18.02.2021).
- 3) Потребление платного контента в России и в мире [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://vc.ru/media/46017-potreblenie-platnogo-kontenta-v-rossii-i-v-mire-kak-izmenyalos-i-chto-ego-zhdet>], свободный. — (дата обращения 05.03.2021).
- 4) Прошедшие события [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [it-events.com], свободный. — (дата обращения 05.03.2021).
- 5) Как «Яндекс» создавал «Алису» [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://vc.ru/future/26878-ya-alice>], свободный. — (дата обращения 05.03.2021).
- 6) Design Efficient In-Database Video Storage Approach by Learning from Performance Evaluation of BLOB [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [https://link.springer.com/chapter/10.1007/978-3-319-27140-8_26], свободный. — (дата обращения 05.03.2021).
- 7) Performance Analysis of Video Storage Based on Clustered NAS Architecture [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [https://link.springer.com/chapter/10.1007/3-540-45453-5_131], свободный. — (дата обращения 05.03.2021).

- 8) Building Big Data Storage Solutions (Data Lakes) for Maximum Flexibility — Электрон. текстовые и фото дан. — Режим доступа: [<https://d1.awsstatic.com/whitepapers/Storage/data-lake-on-aws.pdf>], свободный. — (дата обращения 05.03.2021).
- 9) Исследование проектов CHAOS REPORT [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf], свободный. — (дата обращения 11.03.2021).
- 10) Waterfall — итеративная методология разработки [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://habr.com/ru/post/142023/>], свободный. — (дата обращения 11.03.2021).
- 11) WATERFALL МЕТОДОЛОГИЯ РАЗРАБОТКИ [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://qaevolution.ru/metodologiya-menedzhment/waterfall/>], свободный. — (дата обращения 11.03.2021).
- 12) Что такое Agile-подход и зачем он нужен бизнесу? [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://scrumtrek.ru/blog/agile-scrum/908/chto-takoe-agile-podhod-i-zachem-on-nuzhen-biznesu/>], свободный. — (дата обращения 11.03.2021).
- 13) Постигая Agile. Ценности, принципы, методологии / Эндрю Стеллман, Дженнифер Грин ; пер. с англ. С. Пасерба – 2-е изд. – М. : Манн, Иванов и Фербер, 2018. – 448 с.
- 14) Agile-разработка: преимущества и «подводные камни» [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://rdv-it.ru/company/press-center/blog/agile-razrabotka/>], свободный. — (дата обращения 11.03.2021).

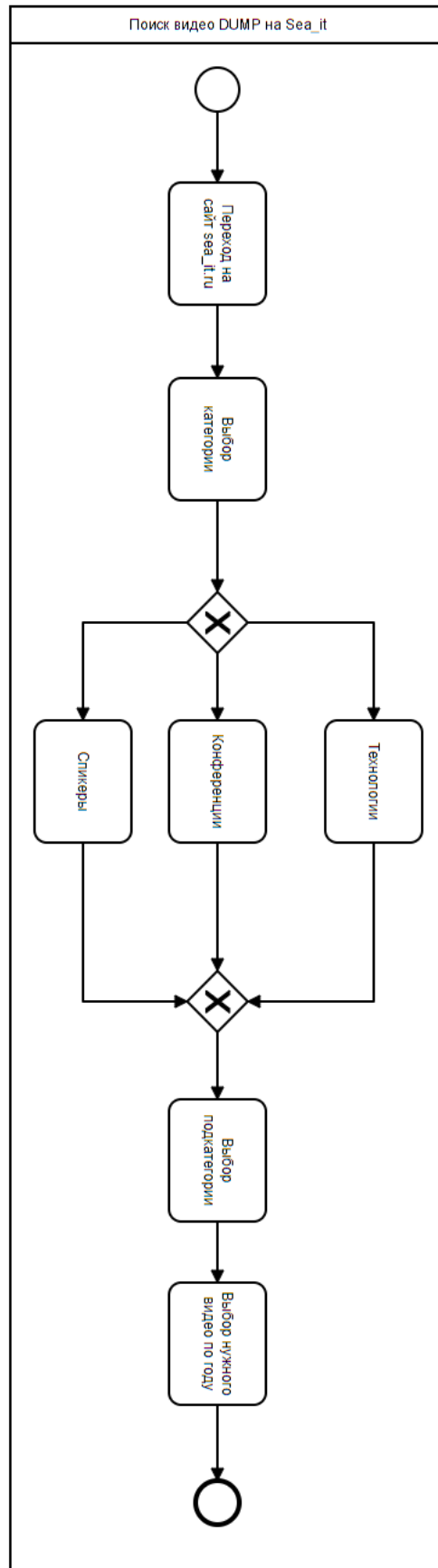
- 15) Что такое Scrum? [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [<https://www.atlassian.com/ru/agile/scrum>], свободный. — (дата обращения 11.03.2021).
- 16) Плюсы и минусы Scrum [Электронный ресурс]. — Электрон. текстовые и фото дан. — Режим доступа: [https://smartfield.ru/what_is_scrum], свободный. — (дата обращения 11.03.2021).
- 17) Знакомимся с Lean: как создать ценность без потерь — Электрон. текстовые и фото дан. — Режим доступа: [https://skillbox.ru/media/management/znakomimsya_s_lean/], свободный. — (дата обращения 11.03.2021).
- 18) Популярные системы управления проектами — Электрон. текстовые и фото дан. — Режим доступа: [<https://www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/>], свободный. — (дата обращения 11.03.2021).
- 19) Методология Kanban: введение — Электрон. текстовые и фото дан. — Режим доступа: [<https://habr.com/ru/post/230725/>], свободный. — (дата обращения 11.03.2021).
- 20) Как Asana упрощает работу с командой, документами и приносит больше профита — Электрон. текстовые и фото дан. — Режим доступа: [<https://habr.com/ru/post/312476/>], свободный. — (дата обращения 11.03.2021).
- 21) Work management resources — Электрон. текстовые и фото дан. — Режим доступа: [<https://asana.com/resources/introduction-to-work-management>], свободный. — (дата обращения 11.03.2021).
- 22) Диаграмма Ганта — Электрон. текстовые и фото дан. — Режим доступа: [<https://asana.com/ru/resources/gantt-chart-basics>], свободный. — (дата обращения 11.03.2021).

- 23) Лучший инструмент разработки для agile-команд — Электрон. текстовые и фото дан. — Режим доступа: [<https://www.atlassian.com/ru/software/jira>], свободный. — (дата обращения 11.03.2021).
- 24) Kanban vs Scrum — how to pick one for your eng team — Электрон. текстовые и фото дан. — Режим доступа: [<https://www.notion.so/blog/kanban-vs-scrum>], свободный. — (дата обращения 11.03.2021).
- 25) Infographic: How Colors Affect Conversions — Электрон. текстовые и фото дан. — Режим доступа: [<https://neilpatel.com/blog/how-colors-affect-conversions/?wide=1/>], свободный. — (дата обращения 11.03.2021).
- 26) Material.io — Электрон. текстовые и фото дан. — Режим доступа: [<https://material.io/components/snackbars/>], свободный. — (дата обращения 18.02.2021).
- 27) Django documentation — Электрон. текстовые и фото дан. — Режим доступа: [<https://docs.djangoproject.com/en/3.2/>], свободный. — (дата обращения 18.02.2021).
- 28) Django REST Framework. Authentication. — Электрон. текстовые и фото дан. — Режим доступа: [<https://www.django-rest-framework.org/api-guide/authentication/>], свободный. — (дата обращения 18.02.2021).
- 29) Yandex Compute Cloud — Электрон. текстовые и фото дан. — Режим доступа: [<https://cloud.yandex.ru/services/compute>], свободный. — (дата обращения 18.02.2021).

ПРИЛОЖЕНИЕ А



ПРИЛОЖЕНИЕ Б



ПРИЛОЖЕНИЕ В

```
@startuml
left to right direction
class confs {
    id : INTEGER
    ..
    name : VARCHAR(255)
    ..
    description: VARCHAR(255)
    ..
    icon: ImageField
    ..
    icon_hash: VARCHAR(255)
}
class technologies {
    id : INTEGER
    ..
    name : VARCHAR(255)
    ..
    parent_name : VARCHAR(255)
    ..
    icon: IMAGEFIELD
    ..
    icon_hash: VARCHAR(255)
}
class speakers {
    id : INTEGER
    ..
    name : VARCHAR(255)
    ..
    tech_id : INTEGER
    ..
    icon: IMAGEFIELD
    ..
    icon_hash: VARCHAR(255)
}
class videos {
    id : INTEGER
    ..
    link : VARCHAR(255)
    ..
    name : VARCHAR(255)
    ..
    tech_id : INTEGER
    ..
    conf_id : INTEGER
    ..
    speaker_id : INTEGER
    ..
    year : DATE
}
technologies::id --> speakers::tech_id
technologies::id --> videos::tech_id
speakers::id --> videos::speaker_id
confs::id --> videos::conf_id
@enduml
```

ПРИЛОЖЕНИЕ Г

#Пример ответа /get?type=confs

```
{
  0:
    {
      "id": 1
      "name": "YaTalks"
      "description": "Яндекс Tanlks - это конференция"
      "icon": /img1242132.jpg
    }
}
```

#Пример ответа /get?type=technologies

```
{
  0:
    {
      "id": 1
      "name": "Python"
      "icon": /img1242132.jpg
    }
  1:
    {
      "id": 2
      "name": "Java"
      "icon": /img124342.jpg
    }
}
```

#Пример ответа /get?type=speakers

```
{
  0:
    {
      "id": 1
      "name": "Волж"
      "tech_id": "Python"
    }
  1:
    {
      "id": 2
      "name": "Иванов"
      "tech_id": "Machine Learning"
    }
}
```