**Реферат**

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Лещенко А.О.*

Провер.

*Сухорукова И.Г.*

Н. контр.

*Рыжанкова А.С.*

Утверд.

Пацей Н.В.

Реферат

Лит.

Листов

1

*74417026, 2020*

У

**Содержание**

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Лещенко А.О.*

Провер.

*Сухорукова И.Г.*

Н. контр.

*Рыжанкова А.С.*

Утверд.

Пацей Н.В.

Реферат

Лит.

Листов

1

*74417026, 2020*

У

**Введение**

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Лещенко А.О.*

Провер.

*Сухорукова И.Г.*

Н. контр.

*Рыжанкова А.С.*

Утверд.

Пацей Н.В.

Введение

Лит.

Листов

1

*74417026, 2020*

У

В настоящее время все больше возрастает количество веб-приложений, причем их характер различен: познавательные, развлекательные, научные, финансовые, медицинские и многие другие. В таком обильном объеме практически каждый может найти то, что для него необходимо.

Также, у каждого человека появляется все больше целей, к которым он может идти на протяжении как от одного часа, так и до конца жизни. С взрослением количество планов и стремлений только увеличивается. С достижением целей в различных сферах появляются специалисты, способные обучить своему умению и других людей.

Процесс нахождения подходящего человека, который обладает определенным навыком на досточно высоком уровне, не всегда прост из-за большого количества образовательных программ с различными возможностями.

Все вышесказанное касается не только людей как таковых, но и компаний, организаций, в которых они работают. Правильная постановка своей активности означает, как правило, повышение по карьерной лестнице.

Веб-приложение, которое будет реализовано в данном дипломном проекте, позволит упростить нахождение ментора для достижения профессиональных целей людей в рамках компании.

Для заполнения доступных навыков предусмотрена роль менеджера, имея которую можно задать уровень для способности. Все навыки могут быть разделены на три уровня: начальный, средний и продвинутый. Данное разбиение будет предсталено и в данном веб-приложении.

Пользователю будет предоставлена возможность просмотреть доступные для него навыки в соответствии с его текущими способностями.

Также, будет реализована функция выбора ментора, который будет курировать работу студента.

В реалиях современного мира, чем выше уровень способностей у человека, тем выше его загруженность, так как востребованность в высококвалифицированных специалистах растет. С учетом данной информации, в разработанном веб-приложении ментор, которому приходит запрос на обучение, может отклонить его.

Чаще всего для повышения квалификации сотрудников, применяется коллективное обучение, что влечет за собой уделение меньшего времени для каждого студента. Для решения данной проблемы будет организована именно частная запись на изучение навыка.

1. **Аналитический обзор литературы и постановка задачи**

ФИО

Подпись

Дата

Лист

1

БГТУ 01.00.ПЗ

Разраб.

*Лещенко А.О.*

Провер.

*Сухорукова И.Г.*

Н. контр.

*Рыжанкова А.С.*

Утверд.

*Пацей Н.В.*

1 Аналитический обзор литературы и постановка задачи

Лит.

Листов

7

*74417026, 2020*

В современном мире веб-приложения являются очень популярными, и с каждым днем их становится все больше.

Образовательная тема также является актуальной. Над обучением студентов различным навыкам работает огромное количество профессионально обученных людей. Сферы навыков очень разнообразны: обучение игре на музыкальных инструментах, обучение программированию, различным техникам рисования, развитию логики, тренировка памяти и многое другое.

Также стоит учесть, что обучение может быть как с реальным человеком, так и автоматизированным, то есть без участия преподавателя. Обучение при общении не с запрограммированным кодом, а с существующим человеком обладает большими преимуществами, так как такой способ позволяет преподавателю с случае недопониманий со стороны студента объяснить изучаемый материал иным образом.

Одним из ключевых моментов в обучении является количество студентов, находящихся в одной группе. Чем больше людей в обучающемся коллективе, тем менее эффективным может стать сам процесс обучения, так как невозможно за небольшой промежуток времени уделить внимание каждому и убедиться в усвоенности пройденного материала. По этой причине индивидуальный подход является более результативным в большей части случаев.

Многие платформы по обучению не предоставляют возможность выбора себе преподавателя, который будет курировать деятельность студента, что важно, так как у всех людей разные подходы к обучению и все индивидуально.

Как правило, специалисты высокого уровня намного загруженнее по времени на проектах в связи со своим опытом, способностью решать более сложные задачи, поэтому предоставление возможности ментору отклонить запрос струдента является важным.

В конце обучения происходит решение о подтверждении либо отклонении освоения навыка, изучаемого студентом на протяжении некоторого времени. Следовательно, должна предполагаться возможность повторной отправки запроса ментору.

Для построения успешного и удобного с точки зрения пользователя приложения, необходимо учитывать пользовательский интерфейс. Для наглядного представления статуса обучения удобно использовать такой интерфейс, который позволит однозначно понять, на каком именно уровне находится тот или иной навык. Для примера, можно использовать несколько колонок, первая из которых будет отображать уже освоенные навыки, вторая включает в себя очередь ожидания подтверждения запросов на обучение и третья может предоставлять список навыков, которые уже находятся в процессе тренинга.

* 1. **Анализ существующих систем**

В настоящее время на рынке представлено большое количество игровых приложений, в которых есть возможность развивать различные навыки с отслеживанием своего прогресса. Что касается корпоративных приложений, то их не так много и в основном они доступны только в рамках компании, но все же можно рассмотреть несколько прототипов веб-приложения, разработанного в данном проекте.

* + 1. **Приложение Skills Score**

Одним из аналогов является веб-приложение «Skills Score». Данный продукт является разработкой компании «iTechArt». Веб-приложение позволяет регистрироваться сотрудникам, работающим в компании. Здесь может быть создан свой профиль, в котором описывается информация о работнике, его фотография, имя и другое.

Несомненным преимуществом данного приложения является то, что пользователю предоставляется возможность выбрать категорию навыков, которую он может изучить, после чего появляется соответствующий список навыков в правой части экрана с указанием названия и уровня, на который можно изучить выбранный пункт. Таким образом, пользователь может выбрать для себя цель. Данная функциональность приложения продемонстрирована на рисунке 1.1.

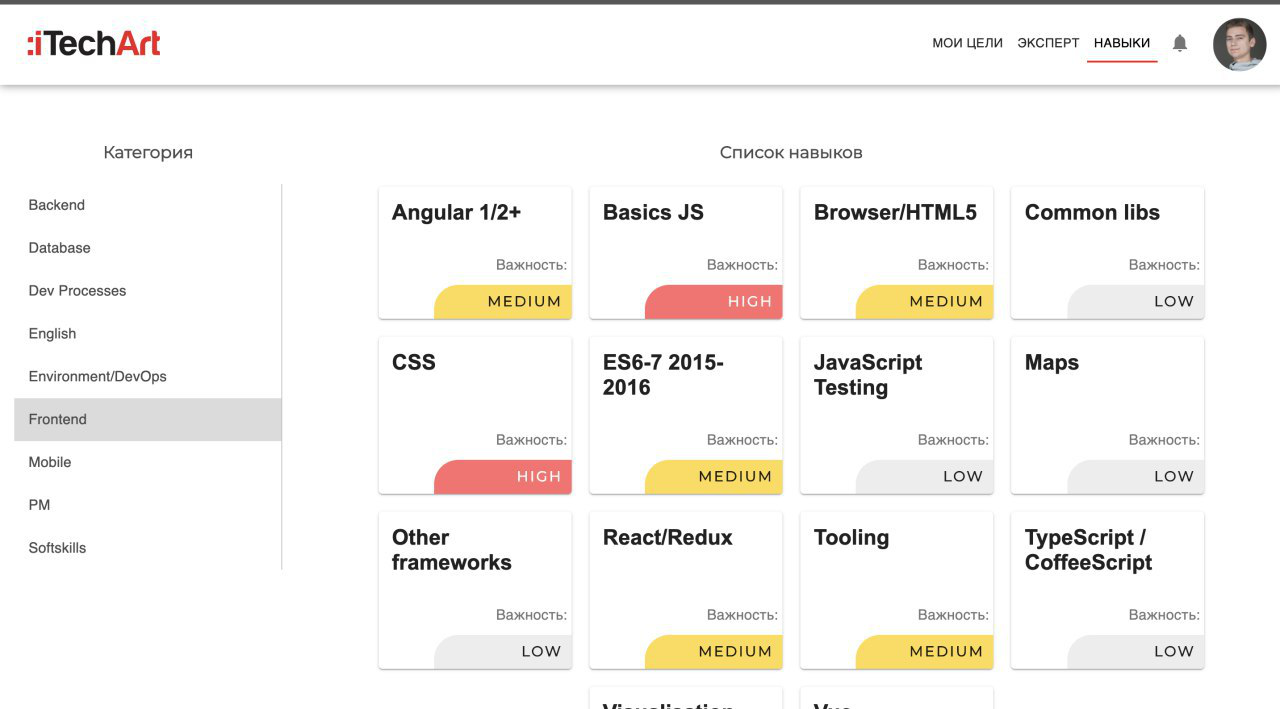


Рисунок 1.1 – Страница навыков веб-приложения «Skills Score»

Еще одним преимуществом данного приложения является наглядность отслеживания целей. На странице «Мои цели» имеется 5 колонок: «Созданные», «Начатые», «В ожидании назначения», «Назначенные» и «Завершенные», каждая из которых имеет свою смысловую нагрузку.

Веб-приложение «Skills Score» позволяет просмотреть текущие цели пользователя с отслеживание статуса цели. Данная функция продемонстрирована на рисунке 1.2.

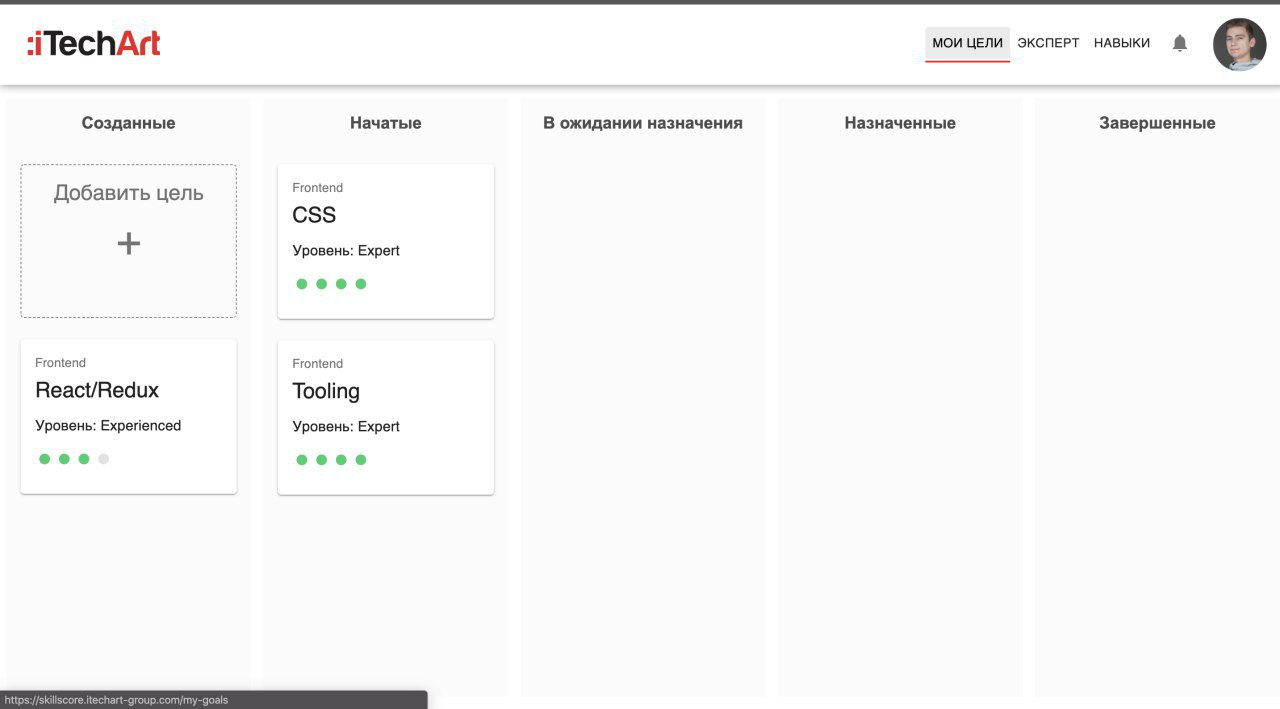


Рисунок 1.2 – Страница целей веб-приложения «Skills Score»

Таким образом, веб-приложение «Skills Score» является очень удобным для пользователя засчет правильно организованного пользовательского интерфейса и возможности фильтрации поиска навыка в соответствии с выбранной категорией.

* + 1. **Приложение Grow**

Вторым прототипом является веб-приложение «Grow», разработанное компанией «EPAM».

Удобным является функция аутентификации с помощью существующего аккаунта почты, что значительно упрощает вход в систему. Также регистрация доступна с помощью учетных записей таких приложений, как «Вконтакте», «Facebook», «Github», «Twitter» и некоторых других.

Также, план изучения цели можно скачать, и полученный файл будет иметь формат «xlsx». В документе отображается название, дата начала, окончания, статус, условие освоения, дата создания, обновления этапа изучения всех этапов выбранного навыка.

Для проверки своих работ можно назначить ментора, который будет курировать и контролировать процесс достижения студентом выбраннной цели. Для этого достаточно ввести существующий адрес электронной почты своего руководителя.

Также, веб-приложение «Grow» имеет приятный графический интерфейс в спокойных тонах.

Данное приложение позволяет добавлять цели с возможностью установки статуса: «Planned», «In Progress», «Done» и «Canceled». Причем в данном веб-приложении предоставлена возможность планирования изучения отдельных тем, связанных с целью. Страница с данной функцией продемонстрирована на рисунке 1.3.

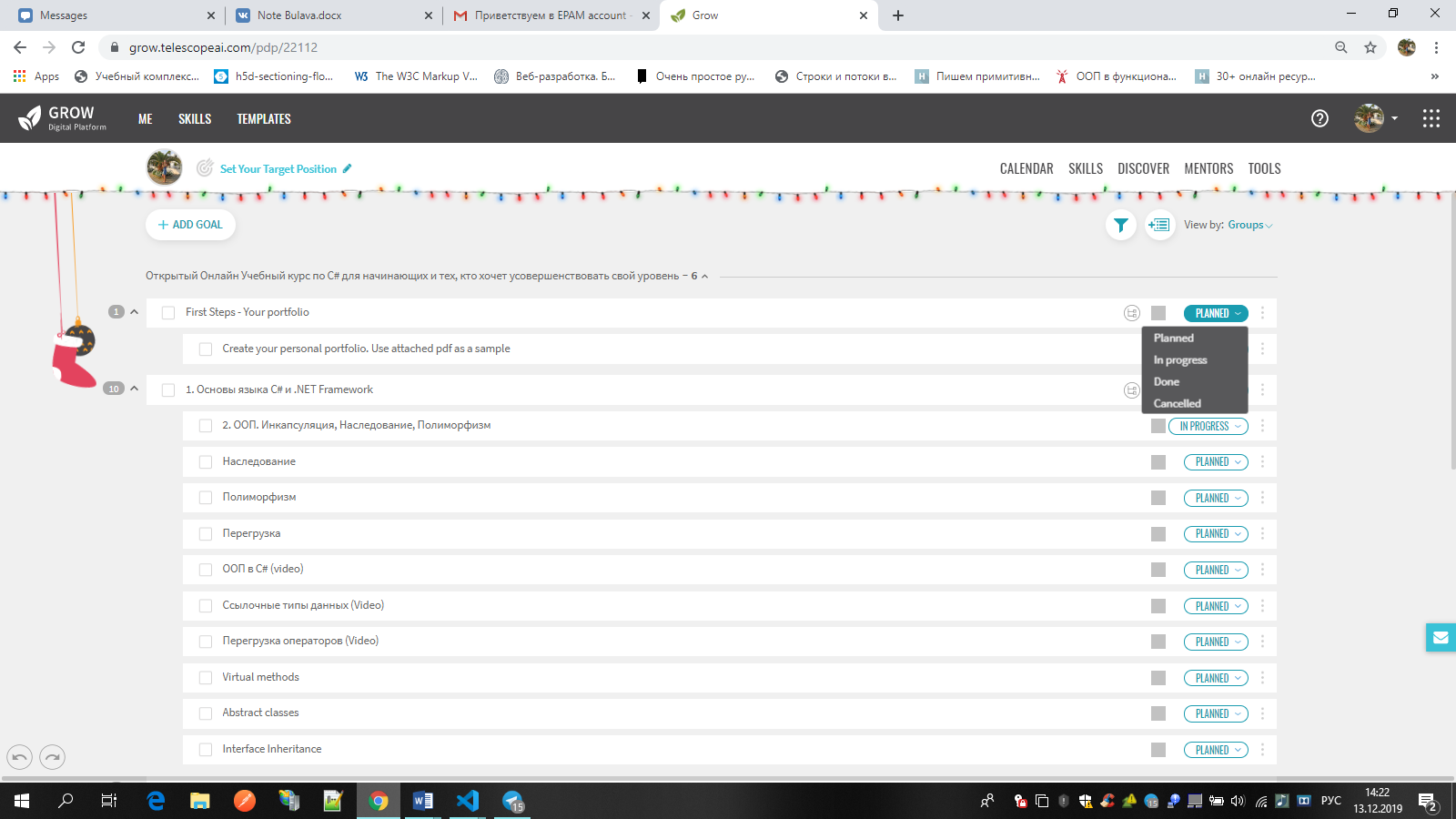


Рисунок 1.3 – Страница целей веб-приложение «Grow»

Также, приложение позволяет выбрать пользователю свою текущую позицию в различных направлениях. Данная функция продемонстрирована на рисунке 1.4.

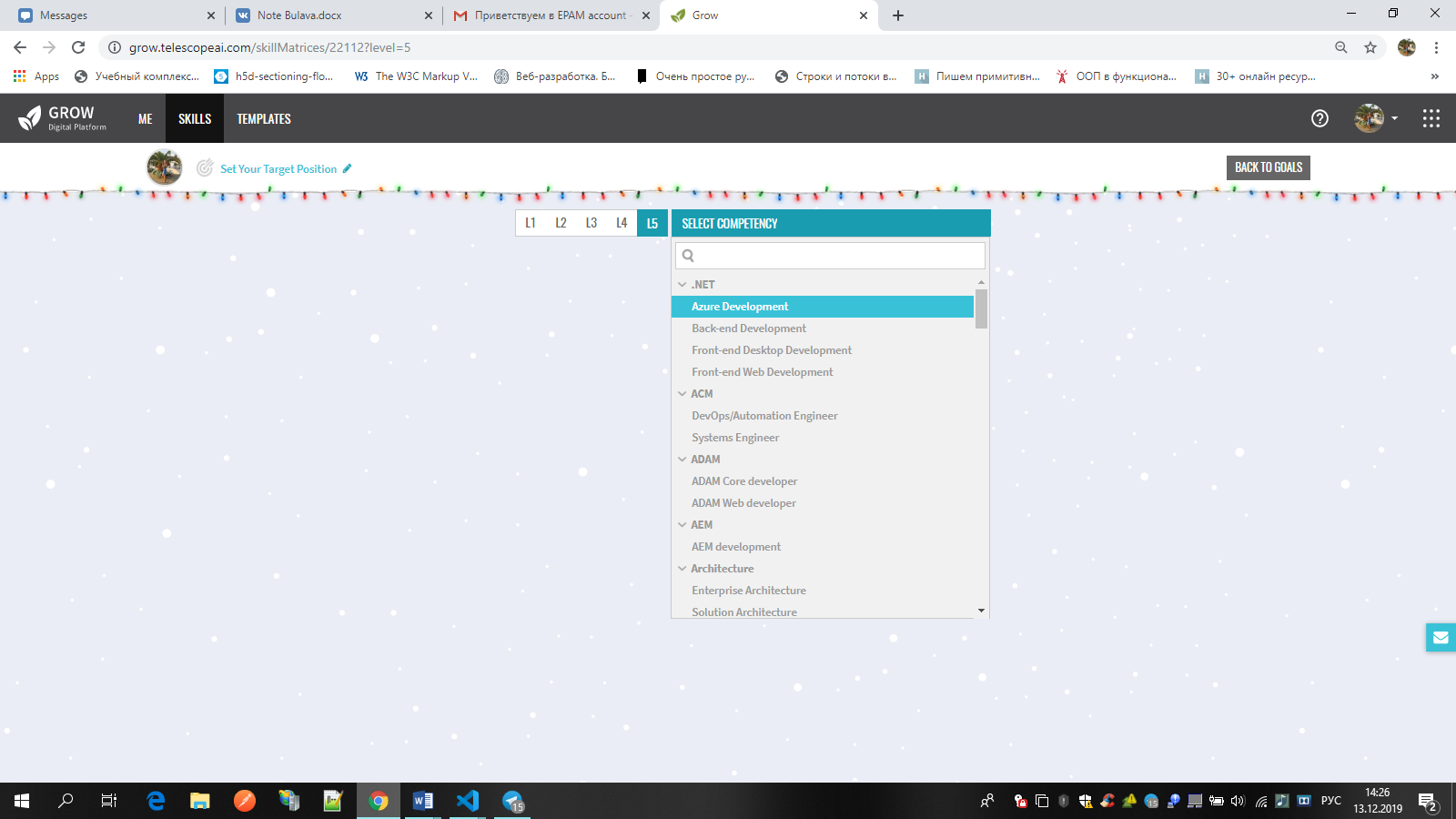


Рисунок 1.4 – Страница навыков веб-приложения «Grow»

Таким образом, оба прототипа предоставляют похожий функционал, где можно заполнить информацию о себе, своих навыках и целях с назначением руководителя.

* 1. **Постановка задачи и выбор технологий для разработки приложения. Основные технические требования к разработке**

Целью данного проекта является создание веб-приложения, которое воссоздает функционал корпоративного продукта, способного организовывать развитие навыков сотрудников с возможностью нахождения и выбора руководителя, который сможет контролировать в последующем всю профессиональную деятельность работника.

Для воплощения этих функций были поставлены следующие задачи:

1. Проанализировать рынок на наличие подобных корпоративных решений с целью изучения их основных возможностей.
2. Разработать UML-диаграммы, которые позволят более подробно и наглядно описать систему.
3. Разработать модель базы данных, что является одной из главных задач всей разработки, так как благодаря именно грамотной структуре хранения информации можно обеспечить долговременную целостность, простоту обслуживания данных и максимальное уменьшение материальных затрат на них.
4. Изучить и научиться грамотно разрабатывать приложение с использованием многоуровневой архитектуры, так как это позволит создать гибкий продукт.
5. Изучить принципы правильной организации серверной части веб-приложения, построенного на Node.js.
6. Изучить применение аутентификации, использующей JWT токены и в последующем разработать промежуточный слой на ее основе.
7. Разработать приложение-клиент, используя Java Script библиотеку React.js для асинхронного рендеринга контента.
   1. **Подходящие технологии для клиентского приложения**

Далее будут рассмотрены различные библиотеки, технологии и подходы, которые могут быть использованы в разработке клиентского приложения.

* + 1. **Язык программирования JavaScript**

JavaScript – это полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах. Его разработал Brendan Eich, сооснователь проекта Mozilla, Mozilla Foundation и Mozilla Corporation.

JavaScript сам по себе довольно компактный, но очень гибкий. Разработчиками написано большое количество инструментов поверх основного языка JavaScript, которые разблокируют огромное количество дополнительных функций с очень небольшим усилием [1].

JavaScript – это легковесный, интерпретируемый или JIT-компилируемый, объектно-ориентированный язык с функциями первого класса. Наиболее широкое применение находит как язык сценариев веб-страниц, но также используется и в других программных продуктах, например, node.js или Apache CouchDB. JavaScript – это прототипно-ориентированный, мультипарадигменный язык с динамической типизацией, который поддерживает объектно-ориентированный, императивный и декларативный (например, функциональное программирование) стили программирования [2].

* + 1. **Библиотека ReactJS**

React – это JavaScript-библиотека для разработки пользовательского интерфейса. Его главная задача – обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты, которые называются компонентами. Каждый компонент – это JavaScript-функция, которая возвращает кусок кода, представляющего фрагмент страницы.

React использует язык программирования, называемый JSX, который похож на HTML, но работает внутри JavaScript, что отличает его от HTML.

Компоненты могут хранить информацию о текущей ситуации. Эта информация называется состоянием (state), она хранится в JS-объекте. Состояние – это инструмент, позволяющий обновлять пользовательский интерфейс, основываясь на событиях.

Компоненты могут «общаться» друг с другом, используя так называемые свойства (props). Свойства – это информация, коллективно используемая родительским компонентом и компонентами-потомками [3].

* + 1. **Стандарт ECMAScript**

ECMAScript — это встраиваемый расширяемый не имеющий средств ввода-вывода язык программирования, используемый в качестве основы для построения других скриптовых языков. Стандартизирован международной организацией ECMA в спецификации ECMA-262. JavaScript является расширением данного языка.

Разработка первой редакции спецификации началась в ноябре 1996 года. Принятие спецификации состоялось в июне 1997 года. В июне 1998 года общим собранием ECMA была принята вторая редакция ECMA-262, соответствующая ISO/IEC 16262. Третья редакция спецификации отличалась от предыдущей введением поддержки регулярных выражений, улучшением поддержки строк, введением новых управляющих конструкций, механизма исключений, форматирования при численном вводе и некоторыми другими изменениями [4].

Переломным моментом в истории EcmaScript является появление ES6, основными особенностями которого являются:

* стрелочные функции;
* классы;
* деструктурирующее присваивание;
* в шаблонных строках можно объявлять строки с помощью обратных кавычек;
* let и const – альтернативы var для объявления переменных;
* итератор и протокол итерации теперь определяют способ перебора любого объекта, а не только массивов. Symbol используется для присвоения итератора к любому объекту;
* добавлены функции-генераторы. Они используют yield для создания последовательности элементов. Функции-генераторы могут использовать yield для делегирования в другую функцию генератора, кроме этого они могут возвращать объект генератора, который реализует оба протокола;
* добавлены промисы.

В связи с вышеперечисленными пунктами можно сделать вывод, что данный стандарт значительно упрощает процесс написания кода и добавляет языку больше гибкости [5].

* 1. **Подходящие технологии для серверного приложения**

В данном разделе будут рассмотрены различные библиотеки, технологии и подходы, которые могут быть использованы в разработке серверного приложения.

* + 1. **Платформа Node.js**

Node.js – это серверная платформа, которая работает на движке Google Chrome – V8, который умеет компилировать JavaScript код в машинный код.

Node.js использует событийно-ориентированную модель и неблокирующую ввод/вывод архитектуру, что делает его легковесным и эффективным. Это не фреймворк и не библиотека, а среда выполнения JavaScript [6].

Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода.

* + 1. **Веб-фреймворк Express**

Express – это один из самых популярных фрейморков Node.js. Он предоставляет такие механизмы, как:

* написание обработчиков запросов HTTP с различными маршрутами;
* интеграция с механизмами рендеринга «view» для генерации ответов путем вставки данных в шаблоны;
* задание общих параметров веб-приложения, таких как порт, используемый для подключения, и расположение шаблонов, используемых для визуализации ответа;
* добавление дополнительной обработки запросов «middleware» в любой точке конвейера обработки запросов [7].
  1. **Вывод по разделу**

Можно выделить разные подходы для построения приложений с использованием различных технологий. В данном дипломном проекте было принято решение использовать Node.js и Express для реализации серверной части в соответствии с многоуровневой архитектурой. Клиентская часть будет выполнена с использованием React.js.

1. **Проектирование**

Проектирование программного обеспечения – процесс создания проекта программного обеспечения. Целью проектирования является определение внутренних свойств системы и детализации её внешних (видимых) свойств на основе выданных заказчиком требований к ПО.

* 1. **Проектирование базы данных**

Одним из ключевых моментов проектирования является создание схемы базы данных. Она включает в себя описание содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных.

Для данного проекта была выбрана БД PostgreSQL. Данная БД является не просто реляционной, а объектно-реляционной, то есть с поддержкой пользовательских объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Это делает PostgreSQL невероятно гибкой и надежной [1].

В проекте была разработана схема базы данных, состоящая из 6 таблиц. Схема представлена в приложении А.

Таблица «Users» хранит в себе информацию о пользователе. В ней находятся следующие поля:

* id – идентификатор, первичный ключ (integer);
* email – e-mail пользователя, уникальный (char);
* password – пароль пользователя (char);
* firstName – имя пользователя (char);
* lastName – фамилия пользователя (char);
* middleName – отчество пользователя (char);
* role – роль, принимает одно из четырех значений: «admin», «manager», «mentor», «student» (char).

Обязательными полями в таблице «Users» являются email и password, а id генерируется средствами базы данных.

Таблица «Mentors» хранит информацию о пользователях, являющихся руководителями. Она содержит следующие поля:

* id – идентификатор, первичный ключ (integer);
* user\_id – внешний ключ на таблицу «Users» поля id (integer);
* status – флаг, отображающий занятость руководителя (boolean).

Отношение таблицы «Users» к таблице «Mentors» – «один-к-одному».

Таблица «Skills» хранит информацию о навыках. В ней находятся следующие поля:

* id – идентификатор, первичный ключ (integer);
* name – название навыка (char);
* level\_name – название уровня, принимает одно из трех значений: «junior», «middle», «senior» (char);
* time\_level – количество месяцев, необходимое на освоение навыка (integer);
* description – описание навыка (char).

В данной таблице пара полей name-level\_name является уникальной.

Таблица «UserSkills» предназначена для организации связи «многие-ко-многим» между сущностями «User» и «Skill», поэтому она содержит следующие поля:

* id – идентификатор, первичный ключ (integer);
* user\_id – внешний ключ на таблицу «Users» поля id (integer);
* skill\_id – внешний ключ на таблицу «Skills» поля id (integer);
* mentor\_id – внешний ключ на таблицу «Mentors» поля id (integer);
* is\_approved\_request – флаг, отображающий подтвержденность запроса на освоение навыка (boolean);
* is\_approved\_skill – флаг, отображающий подтвержденность прохождения навыка (boolean).

В данной таблице группа полей user\_id-skill\_id-mentor\_id является уникальной.

Таблица «Students» хранит информацию о студентах и содержит следующие поля:

* id – идентификатор, первичный ключ (integer);
* userskill\_id – внешний ключ на таблицу «UserSkills» поля id (integer);
* start\_time\_mentor – дата и время, когда у студента появился руководитель (timestamp with time zone);
* finish\_time\_mentor – дата и время, когда студент закончил освоение навыка (timestamp with time zone).

Таблица «Goals» содержит информацию о целях пользователя и представлена следующими полями:

* id – идентификатор, первичный ключ (integer);
* student\_id – внешний ключ на таблицу «Students» поля id (integer);
* date\_review – дата, когда будет решаться, подтвердил ли студент выбранный навык (timestamp with time zone).
  1. **Проектирование системы**

Для описания функционального назначения системы была разработана UML Use-Case диаграмма. Для начала, необходимо обозначить нотации, характерные для данного вида диаграммы и продемонстрированные в данном проекте: актеры, прецеденты, подсистема.

Актеры – множество ролей, которые пользователи могут осуществлять в процессе работы, причем в качестве пользователя может выступать как человек, так и система.

Прецедент – нотация диаграммы, которая отвечает на вопрос «что можно сделать?».

Подсистема – прямоугольник, в котором располагаются все прецеденты [2].

Построенная схема представлена в приложении Б. На ней можно выделить четырех актеров: «Пользователь», «Менеджер», «Студент» и «Руководитель».

«Пользователь» – это актер, от которого наследуются остальные, и ему доступен только один прецедент: «Войти в систему».

Прецеденты для актера «Менеджер» следующие:

* просмотреть таблицу «Skills»:
  1. добавить навык;
  2. изменить навык;
  3. удалить навык;
* просмотреть таблицу «Mentors»:
  1. добавить руководителя;
  2. просмотреть навыки руководителя;
  3. удалить руководителя.

Прецеденты для актера «Студент»:

* просмотреть текущие навыки;
* просмотреть доступные навыки:
  1. выбрать ментора:
     1. отрпавить запрос на изучение навыка;
* просмотреть очередь запросов на изучение навыка:
  1. отменить запрос;
* просмотреть подтвержденные запросы.

Прецеденты для актера «Руководитель»:

* просмотреть запросы на изучение навыка:
  1. принять запрос;
  2. отклонить запрос;
* просмотреть текущих студентов:
  1. подтвердить навык.

В целом, диаграмма вариантов использования преследует следующие цели:

* + определение общих границ и контекста моделируемой предметной области на начальных этапах проектирования;
  + формулировка общих требований к функционированию поведения информационной системы;
  + разработка исходной концептуальной модели системы для ее последующей детализации в форме логических и физических моделей;
  + подготовка исходной документации для взаимодействующих разработчиков системы с ее заказчиками и пользователями [3].

В настоящее время тема безопасности очень важна для всех пользователей приложений, независимо от платформы и вида приложения.

Для обеспечения ограничения доступа к веб-приложению неавторизованных пользователей в данном проекте используется аутентификация.

Аутентификация – это процедура проверки подлинности введенных пользователем данных. С данным понятием также связаны такие термины, как идентификация и авторизация.

Идентификация – это заявление пользователя о себе, у каждого пользователя есть свой идентификатор.

Авторизация – предоставление пользователю прав на выполнение определенных действий в системе.

Выделяют различные виды аутентификации: Forms, Basic, Digest, OAuth, OpenID и некоторые другие.

В данном проекте продемонстрирована аутентификация на основе JWT-токенов.

**Заключение**

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Лещенко А.О.*

Провер.

*Сухорукова И.Г.*

Н. контр.

*Рыжанкова А.С.*

Утверд.

Пацей Н.В.

Заключение

Лит.

Листов

1

*74417026, 2020*

У

**Список использованнх источников**

ФИО

Подпись

Дата

Лист

1

БГТУ 00.00.ПЗ

Разраб.

*Лещенко А.О.*

Провер.

*Сухорукова И.Г.*

Н. контр.

*Рыжанкова А.С.*

Утверд.

Пацей Н.В.

Список использованных источников

Лит.

Листов

1

*74417026, 2020*

У

1. Основы JavaScript [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/JavaScript\_basics. – Дата доступа: 06.05.2020.
2. JavaScript [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/ru/docs/Web/JavaScript. – Дата доступа: 06.05.2020.
3. Основы React [Электронный ресурс] – Режим доступа: https://habr.com/ru/company/ruvds/blog/343022/. – Дата доступа: 07.05.2020.
4. ECMAScript [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/ECMAScript. – Дата доступа: 07.05.2020.
5. JavaScript brief history and ECMAScript [Электронный ресурс] – Режим доступа: https://medium.com/@madasamy/javascript-brief-history-and-ecmascript-es6-es7-es8-features-673973394df4. – Дата доступа: 07.05.2020.
6. Все, что вам нужно знать о Node.js [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/460661/. – Дата доступа: 08.05.2020.
7. Express/Node introduction [Электронный ресурс] – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\_Nodejs. – Дата доступа: 08.05.2020.