**1.** ОБЗОР ЛИТЕРАТУРЫ

1.1 Обзор используемых технологий

В данном разделе выберем технологии, которые будут использованы для реализации настоящего проекта.

1.1.1 Выбор front-end технологий

На сегодняшний день в области разработки web-приложений существует очень распространенная тенденция: SPA(Single Page Application) или одностраничное приложение. По сути сервер осуществляет рендеринг html-страницы лишь при первом обращении. При последующих запросах сервер и клиент осуществляют обмен данными (например в формате JSON), а непостредственно рендеринг страницы производится на стороне клиента. У тенденции single page application есть два неоспоримых преимущества: экономия сетевого трафика и уменьшение нагрузки на серверную часть приложения. Поэтому в настоящем обзоре будут рассмотрены технологии упрощающие реализацию концепции single page application.

Сюда

1.1.1.1 AngularJS

AngularJS является мощным структурным фреймворком для динамических web-приложений. Главная особенность AngularJS заключается в использовании шаблона проектирования MVW (Model-View-Whatever) и расширение традиционного HTML синтаксиса за счет директив. Другая отличительная черта фреймворка - это двусторонняя привязка данных для автоматической синхронизации модели и представления. Кроме того, AngularJS поддерживает внедрение зависимостей и предлагает множество других полезных функций, таких как маршрутизация, Ajax, валидация форм, шаблоны и анимация.  
AngularJS имеет достаточно низкий порог вхождения благодаря обширной документации и многочисленным учебным материалам, как из официальных источников, так и из созданных при поддержке сообщества. Разработчик AngularJS – компания Google, предоставляет официальное руководство по стилю написания кода на данном фреймворке.

1.1.1.2 ReactJS

ReactJS ­ это JavaScript фреймворк разработанный компанией Facebook, который используется для построения пользовательского интерфейса. Одной из его отличительных особенностей является возможность использовать JSX язык программирования с близким к HTML синтаксисом, который компилируется в JavaScript. Разработчики могут добиваться высокой производительности приложений с помощью Virtual DOM. C ReactJS возможно создавать изоморфные приложения, в которых данные могут подгружаться с сервера в асинхронном режиме, без полной перезагрузки страницы. Созданные компоненты могут быть с легкостью изменены и использованы заново в новых проектах. Высокий процент переиспользования кода повышает покрываемость тестами, что, в свою очередь, приводит к более высокому уровню контроля качества.

1.1.1.3 Резюме

Резюмируя всё вышеизложенное был сделан выбор в пользу фреймворка AngularJS, поскольку он имеет достаточно низкий порог вхождения благодаря обширной документации и многочисленным учебным материалам, как из официальных источников, так и из созданных при поддержке сообщества. В связи с ограниченным сроком, на реализацию настоящего проекта, этот фактор приобрел ключевое значение.

1.1.2 Выбор back-end технологий

В связи со знакомством с языкам Java и имеющимся опытом работы с данным языком программирования, именно этот язык был выбран в качестве реализации серверной части настоящего web-приложения. Однако в Java существует множество технологий упрощающих разработку серверной части web-приложений, рассмотрим некоторые из них и выберем наиболее подходящие.

1.1.2.1 Spring Framework

Spring Framework ­- это универсальный Java фреймворк с открытым исходным кодом. Spring состоит из множества модулей, которые разработчик может подключать к своему приложению по необходимости. Подавляющее большинство этих подулей может работать независимо друг от друга. По умолчанию Spring предоставляет две основных возможности: Inversion of Control (каждый объект получает свои зависимости, а не создает их самостоятельно), Dependency Injection (внедрение зависимостей). Spring имеет низкий порог вхождения благодаря обширной документации и большому количеству различных статей в сети интернет.

1.1.2.2 JSF

Фреймворк JSF (Java Server Faces) был разработан компанией Oracle в рамках спецификации J2EE. Данный фреймворк не совсем подходит для быстрой разработки web-приложений, однако он легок в использовании благодаря наличию качественной документации и отсутствию внешних зависимостей, то есть при использовании данного фреймворка мы не выходим за рамки спецификации J2EE. JSF славится своей богатой инфраструктурой и различными модулями, которые подходят на все случаи жизни: разработка пользовательского интерфейса, взаимодействие с базами данных и др. Главным недостатком данного фреймворка является сложность его внутреннего устройства.

1.1.2.3 Grails

Данный фреймворк, в отличие от предыдущих, написан не на языке Java, а на языке Groovy, однако он является полностью совместимым с языком программирования Java. Вообще по своей сути Grails можно считать не фрейврком, а своеобразной точком для интерграции других фреймворков. Это позволяет Grails взаимодействовать с многими фреймворками, например, Spring, Hibernate, Struts 2 и другие. На ряду с вышеупомянутыми фреймворками Grails содержит в себе различные плагины для интеграции к контейнерами сервлетов, базами данных и IDE.

1.1.2.4 Резюме

Среди рассмотренных фреймворков в качестве инструмента для разработки настоящего проекта будет использован Spring Framework, поскольку на сегодняшний день этот фреймворк является самым популярным среди Java разработчиков. Также Spring содержит набор стандартных конфигураций Spring Boot, воспользовавшись которыми отпадает необходимость настройки проекта вручную.

1.1.3 Выбор СУБД

В настоящем пункте будет выбрана база данных для хранения всей необходимой информации. В настоящее время всё большую популярность набирают NoSQL базы данных, то есть данные хранятся не в виде таблиц, а, например, в виде графа.

Однако, в связи с имеющимся опытом работы выбор пал всё же в сторону SQL баз данных.

1.1.3.1 SQLite

SQLite по своей сути является файловой базой данных: все таблицы хранятся в виде файлов на локальном компьютере. Когда приложение использует SQLite, их связь производится с помощью функциональных и прямых вызовов файлов, содержащих данные (например, баз данных SQLite), а не интерфейса, что повышает скорость и производительность операций. В таблице 1 приведены все типы данных, которые возможно хранить в СУБД SQLite.

Таблица 1. Типы данных SQLite

|  |  |
| --- | --- |
| Название типа | Краткое описание |
| NULL | Пустое значение в таблице базы |
| INTEGER | Целочисленное значение, хранящееся в 1, 2, 3, 4, 6 или 8 байтах, в зависимости от величины самого значения |
| REAL | Числовое значение с плавающей точкой. Хранится в формате 8-байтного числа IEEE с плавающей точкой |
| TEXT | Значение строки текста. Хранится с использованием кодировки базы данных (UTF-8, UTF-16BE или UTF-16LE) |
| BLOB | Значение бинарных данных, хранящихся точно в том же виде, в каком были введены |

### Преимущества SQLite:

- файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины;

­- используемые стандарты - хотя может показаться, что эта СУБД примитивная, но она использует SQL. Некоторые особенности опущенны (RIGHT OUTER JOIN или FOR EACH STATEMENT), но основные все-таки поддерживаются;

­ - в процессе разработки приложений часто появляется необходимость масштабирования. SQLite предлагает всё что необходимо для этих целей, так как состоит всего из одного файла и библиотеки написанной на языке C.

Недостатки SQLite:

### ­ - отсутствие системы пользователей - более крупные СУБД включают в свой состав системы управления правами доступа пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях;

### ­ - отсутствие возможности увеличения производительности - опять, исходя из проектирования, довольно сложно выжать что-то более производительное из этой СУБД.

### 1.1.3.2 MySQL

### MySQL в настоящее время является одной из самых популярных СУБД, используемых при разработке высоконагруженных приложений. Одновременно MySQL проста в использовании. Для нее существует огромное количество документации и различных статей, помогающих начать работу с данной СУБД. В таблице 2 приведены все типы данных, которые можно хранить в СУБД MySQL.

### Таблица 2. Типы данных MySQL

|  |  |
| --- | --- |
| Название типа | Краткое описание |
| TINYINT | очень малые целочисленные значения |
| SMALLINT | малые целочисленные значения |
| MEDIUMINT | средние целочисленные значения |
| INTEGER | стандартные целочисленные значения |
| BIGINT | большие целочисленные значения |
| DOUBLE, BOUBLE PRECISION, REAL | Стандартные значения с плавающей точкой. Всегда знаковые |
| DECIMAL, NUMERIC | распакованное значение с плавающей точкой, всегда знаковое |
| DATE | дата |
| DATETIME | дата и время в одном значении |
| TIMESTAMP | временная отметка timestamp |
| TIME | время |
| YEAR | год, 2 или 4 числа (4 - по-умолчанию) |
| CHAR | строковое значение фиксированной длины, справа всегда добавляются пробелы до указанной длины при сортировке |
| VARCHAR | строковое значение переменной длины |
| TINYBLOB, TINYTEXT | значение типа BLOB или TEXT, 255 (2^8 - 1) символов - максимальныя длина |
| BLOB, TEXT | значение типа BLOB или TEXT, 65535 (2^16 - 1) символов - максимальныя длина |
| MEDIUMBLOB, MEDIUMTEXT | значение типа BLOB или TEXT, 16777215 (2^24 - 1) символов - максимальныя длина |
| LONGBLOB, LONGTEXT | значение типа BLOB или TEXT, 4294967296 (2^32 - 1) символов - максимальныя длина |
| ENUM | перечисление |
| SET | множество |
| FLOAT | маленькие значения с плавающей точкой (точность до одного значения после точки). Всегда знаковые значения |

### Преимущества MySQL:

- Простота в работе - установить MySQL довольно просто. Дополнительные приложения, например GUI, позволяет довольно легко работать с БД,

- Богатый функционал - MySQL поддерживает большинство функционала SQL,

- Безопасность - большое количество функций обеспечивающих безопасность, которые поддерживается по умолчанию,

- Масштабируемость - MySQL легко работает с большими объемами данных и легко масштабируется,

- Скорость - упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность.

### Недостатки MySQL:

### - Известные ограничения - по задумке в MySQL заложены некоторые ограничения функционала, которые иногда необходимы в особо требовательных приложениях,

### - Проблемы с надежностью - из-за некоторых способов обработки данных MySQL (связи, транзакции, аудиты) иногда уступает другим СУБД по надежности,

### - Медленная разработка - Хотя MySQL технически открытое ПО, существуют жалобы на процесс разработки. Стоит заметить, что существуют другие довольно успешные СУБД созданные на базе MySQL, например MariaDB.

1.1.3.3 PostgreSQL

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность (Atomicity, Consistency, Isolation, Durability (ACID).) Благодаря мощным технологиям Postgre очень производительна. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID. PostgreSQL очень легко расширять своими процедурами, которые называются хранимые процедуры. Эти функции упрощают использование постоянно повторяемых операций. В таблице 3 представлены все типы данных, которые можно хранить в СУБД PostgreSQL.

Таблица 3. Типы данных PostgreSQL

|  |  |
| --- | --- |
| Название типа | Краткое описание |
| bigint | знаковое 8-ми битное целочисленное значение |
| bigserial | автоматически инкрементируемое 8-ми битное целочисленное значение |
| bit[(n)] | строка постоянной длины |
| bit varying [(n)] | строка переменной длины |
| boolean | булево значение (true/false) |
| box | прямоугольник на плоскости |
| bytea | бинарные данные (массив байтов) |
| character varying [(n)] | строковое значение переменной длины |
| character [(n)] | строковое значение постоянной длины |
| cidr | IPv4/IPv6 сетевой адрес |
| circle | круг на плоскости |
| date | календарная дата (год, месяц, день) |
| double precision | число с плавающей точкой двойной точности (8 байт) |
| inet | IPv4/IPv6 адрес хоста |
| integer | знаковое 4-ех байтовое целочисленное значение |
| interval [fields][(p)] | отрезок времени |
| line | бесконечная прямая на плоскости |
| lseg | отрезок на плоскости |
| macaddr | MAC адрес |
| mony | валютное значение |
| numeric [(p, s)] | точное численное значение с выбранной точностью |
| path | геометрическая кривая на плоскости |
| point | геометрическая точка на плоскости |
| polygony | многоугольник на плоскости |
| real | число с плавающей точкой одинарной точности (4 байта) |
| smallint | знаковое целочисленное значение (4 байта) |
| serial | автоматические инкрементируемое целочисленное значение (4 байта) |
| text | строковое значение переменной длины |
| time [(p)] [without time zone] | время суток (без часового пояса) |
| time [(p)] with time zone | время суток (включая часовой пояс) |
| timestamp [(p)] [without time zone] | дата и время (без часового пояса) |
| timestamp [(p)] with time zone | дата и время (с часовым поясом) |
| tsquery | текстовый поисковый запрос |
| tsvector | документ текстового поиска |
| txid\_snapshot | пользовательский снимок транзакции с ID |
| uuid | универсальный уникальный идентификатор |

Достоинства PostgreSQL:

- открытое ПО соответствующее стандарту SQL - PostgreSQL - бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой;

- большое сообщество - существует довольно большое сообщество в котором вы запросто найдёте ответы на свои вопросы;

- большое количество дополнений - несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими;

- расширения - существует возможность расширения функционала за счет сохранения своих процедур;

- объектность - PostrgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого.

### Недостатки PostgreSQL:

- производительность - при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL;

- популярность - по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество;

- хостинг - в силу выше перечисленных факторов иногда довольно сложно найти хостинг с поддержкой этой СУБД.

1.1.3.3 Резюме

В настоящем проекте было решено использовать базу данных MySQL. Именно эта СУБД будет использована, потому что она является одной из самых популярных среди всех ныне существующих реляционных баз данных, а так же в связи с имеющимся опытом разработки на данной СУБД.

**1.2** Обзор аналогов

Как было упомянуто ранее, поскольку методологии с использованием Kanban досок сильно распространены в настоящее время, сооответственно и аналогов у настоящего программного продукта большое количество. Рассмотрим некоторые из них более детально.

1.2.1 Trello

Trello представляет собой доску с задачами, которые распределены по колонкам (см. рис. 1). В каждой колонке содержится задача принадлежащая той или иной группе.

Каждая задача представлена в виде карточки. С карточками можно проводить большое количество манипуляций (см. рис. 3): комментирование, прикрепление файлов, назначение меток и др.

### 

Рисунок 3 – ­ управление задачей.

В Trello для каждого проекта создается своя рабочая группа (см. рис. 4). Все члены данной группы имеют возможность просматривать в режиме реального времени состояние задач касающихся проекта. Также возможно получать уведомления офлайн: по электронной почте.



Рисунок 4 –­ рабочая группа проекта

В общем случае trello представляет собой web-приложение, однако имеется и приложение для мобильных устройств под управлением операционной системы IOS.

По мнению некоторых разработчиков Trello не годится для разработки крупномасштабных проектов, а больше подходит для различого рода стартапов или просто как место где можно быстро набросать имеющиеся идеи. Использование всех версий Trello в настоящий момент полностью бесплатно.

1.2.2 Kanban flow.

Данный сервис являетс абсолютно бесплатным и позволяет создать несколько досок, при необходимости работы сразу над несколькими проектами.

Изначально пользователю предоставляется доска с четырьмя колонками (см. рис. 2), в каждой колонке задачи с определенным статусом:

- новые задачи;

- задачи на сегодня;

- задачи, которые выполняются;

- выполненные задачи.

При добавлении новой задачи в проект можно указывать: время, которое потребуется для решения задачи, комментарий к задаче и человека, который будет эту задачу решать (см. рис. 5). При работе над задачей можно указать фактическое время ее выполнения.



Рисунок 5 — Создание новой задачи kanban flow.

Смена статуса задачи осуществляетс методом drag and drop, а при нажатиии на задачу правой кнопкой мыши, появляется контекстное меню со списком наиболее популярных действий над задачей.

Данная доска является полностью настраиваемой, при необходимости пользователь может изменить порядок колонок, изменить название колонок, добавить или удалить колонки (см. рис. 6).

Также существует возможность добавления новых людей к проекту и коллективная работа над ним. Приглашенному пользователю придет уведомление на электронную почту.



Рисунок 6 –­ Редактирование колонок Kanban flow.

В итоге можно сказать, что доска Kanban flow довольно функциональна и удобна, одним из недостатков, для некоторых пользователей, является отсутствие мобильного приложения Kanban flow.

**1.3** Резюме

Резюмируя всё вышесказанное можно сказать, что настоящий дипломный проект будет состоять из двух частей: серверная часть, и клиентская (браузерная) часть. Серверная часть будет написана на языке программиования Java, с использованием фреймворка Spring. Клиентская часть будет написана с использованием фреймворка AngularJS. В качестве СУБД была выбрана MySQL.