**1** ОБЗОР ЛИТЕРАТУРЫ

**1.1** Обзор используемых технологий

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

**1.1.1** Выбор технологий клиентской части приложения

Пропущено, что сайты могут быть представлены так-то и так-то. А затем уже переходим к тому, что популярно

На сегодняшний день в области разработки веб-приложений существует 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 непосредственно рендеринг страницы производится на стороне клиента. У тенденции SPA есть два неоспоримых преимущества: экономия сетевого трафика и уменьшение нагрузки на серверную часть приложения. Поэтому в настоящем обзоре будут рассмотрены технологии, упрощающие разработку SPA приложений.

**1.1.1.1** Angular

Angular является мощным структурным фреймворком для динамических веб-приложений. Главная особенность Angular заключается в использовании шаблона проектирования MVW (Model-View-Whatever) (Модель-Отображение-Что угодно) и расширение традиционного HTML синтаксиса директивами. Другая отличительная черта фреймворка — это автоматическая синхронизации модели и представления. Кроме того, Angular поддерживает внедрение зависимостей и предлагает множество других полезных функций, среди которых можно выделить маршрутизацию, асинхронные запросы, валидацию форм, шаблоны и анимацию.  
Angular имеет достаточно низкий порог вхождения благодаря обширной документации и многочисленным учебным материалам, как из официальных источников, так и созданным крупным сообществом разработчиков. Разработчик Angular – компания Google, предоставляет официальное руководство по стилю написания кода на данном фреймворке. Разработка на данном фреймфорке может осуществляться на двух языках программирования: JavaScript и TypeScript.

**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 транслируется в JavaScript. Разработчики могут добиваться высокой производительности приложений с помощью Virtual DOM (Объектная модель документа). 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 повторно использованы в новых проектах. Высокий процент переиспользования кода повышает покрываемость тестами, что, в свою очередь, приводит к более высокому уровню контроля качества.

Из-за низкого порога вхождения и наличия обширной документации выбор был сделан в пользу фреймворка Angular. В связи с ограниченным сроком, на реализацию настоящего проекта, этоти факторы приобрели ключевое значение.

**1.1.2** Выбор технологий для разработки веб-сервиса

Поскольку для языка Java существует множество технологий, упрощающих разработку серверной части веб-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 не происходит выход за рамки спецификации J2EE. JSF славится своей богатой инфраструктурой и различными модулями, которые подходят на для решения широкого круга задач: 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 платформой для интерграции иных фреймворков. Это позволяет Grails взаимодействовать со многими фреймворками, например, Spring, Hibernate, Struts 2 и другими. Вместе с вышеупомянутыми фреймворками Grails содержит в себе различные плагины для интеграции с контейнерами сервлетов, базами данных и IDE.

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

Выберем базу 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 настоящей предметной области прослеживается явное наличие объектов(задача, проект, пользователь и др), что явно хорошо проецируется на реляционные базы данных. В связи с этим фактом, а также в связи с имеющимся опытом работы с реляционными СУБД, выбор пал в их сторону.

**1.1.3.1** SQLite

SQLite по своей сути является файловой базой данных: все таблицы хранятся в виде файла 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.

Типы данных, которые возможно хранить в базе данных SQLite: NULL, INTEGER, REAL(исловое 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 возможности SQL в ней недоступны (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;

### отсутствие возможности увеличения производительности;

### вероятность повреждения данных при асинхронном доступе к ним.

### **1.1.3.2** MySQL

### MySQL в настоящее время является одной из самых популярных СУБД, используемых при разработке высоконагруженных приложений. Для нее существует огромное количество документации и различных статей, помогающих начать работу с данной СУБД.

### MySQL поддерживает большое количество типов данных: TINYINT(очень малые целочисленные значения), SMALLINT(малые целочисленные значения), MEDIUMINT(средние целочисленные значения), INTEGER, BIGINT(большие целочисленные значения), DOUBLE, DECIMAL(распакованное значение с плавающей точкой, всегда знаковое), DATE(дата), DATETIME(дата и время в одном значении), TIMESTAMP(временная отметка timestamp), TIME, YEAR (год, 2 или 4 числа), CHAR, VARCHAR(строковое значение переменной длины), FLOAT, SET(множество), ENUM(перечисление), BLOB(значение типа BLOB 65535 (2^16 - 1) символов - максимальная длина), TINYBLOB, MEDIUMBLOB, LONGBLOB.

### Преимущества MySQL:

* простота использования;
* поддержка большинства функций языка SQL;
* реализовано большое количество средств, обеспечивающих безопасность;
* высокая производительность.

### Недостатки MySQL:

### не все возможности SQL реализованы;

### невысокая надежность, связанная со способами обработки данных;

### медленная разработка: новые версии СУБД выходят с большой задержкой.

**1.1.3.3** PostgreSQL

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка транзакций. Акцент разработчиков был уделен производительности СУБД. Параллельность достигается не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом. PostgreSQL легко дополняется хранимыми процедурами. В таблице 3 представлены все типы данных, которые можно хранить в СУБД PostgreSQL.

Типы данных поддерживаемые PostgreSQL: 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, box(прямоугольник на плоскости), bytea(массив байт), character varying(строковое значение переменной длины), character(строковое значение постоянной длины), cidr(IPv4/IPv6 сетевой адрес), circle(круг на плоскости), date(календарная дата), double, inet (IPv4/IPv6 адрес хоста), integer, interval [fields][(p)](отрезок времени), 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(прямоугольик на плоскости), real(число с плавающей точкой одинарной точности), smallint(знаковое целочисленное значение), serial(автоматические инкрементируемое целочисленное значение), 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;
* существование большого сообщества, что помогает найти решение некоторых проблем;
* наличие большого количества дополнений, которые позволяют разрабатывать данные для этой СУБД и управлять ими;
* возможность расширения функционала за счет вставки хранимых процедур;
* реализация объектно-ориентированной модели.

### Недостатки PostgreSQL:

* слабая производительность (особенно при чтении);
* низкая популярность;
* мало хостингов поддерживает данную СУБД.

В настоящем проекте будет использоваться база 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. Также выбор в сторону MySQL пал, потому что данная база является бесплатной, кроссплатформенной и хорошо подходит для хранения структурированных данных.

**1.2** Обзор аналогов

Как было упомянуто ранее, поскольку методологии с использованием Kanban досок сильно распространены в настоящее время, соответственно 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 1.3 – Экран управления 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 1.4 –­ Экран рабочей группы 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 является 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 1.5 — Создание новой задачи в kanban flow.

Смена статуса задачи осуществляется методом перетаскивания, а при нажатии на задачу правой кнопкой мыши, появляется контекстное меню со списком наиболее некоторых 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 1.6 –­ 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** Выводы

Резюмируя всё вышесказанное можно сказать, что настоящий дипломный проект будет состоять из серверной и клиентской частей. Серверная часть будет написана на языке программирования Java, с использованием фреймворка Spring. Клиентская часть будет написана с использованием фреймворка AngularJS на языке JavaScript. В качестве СУБД была выбрана MySQL.