## ГЛАВА 1. ОСНОВНЫЕ ИНСТРУМЕНТЫ РАЗБРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ

## Применение системы прототипирования Invision

InVision – платформа для разработки макетов интерфейсов. Данная платформа предоставляет возможность разработки программного продукта, в котором можно создавать дизайнерские макеты и просматривать работу макета в программе.

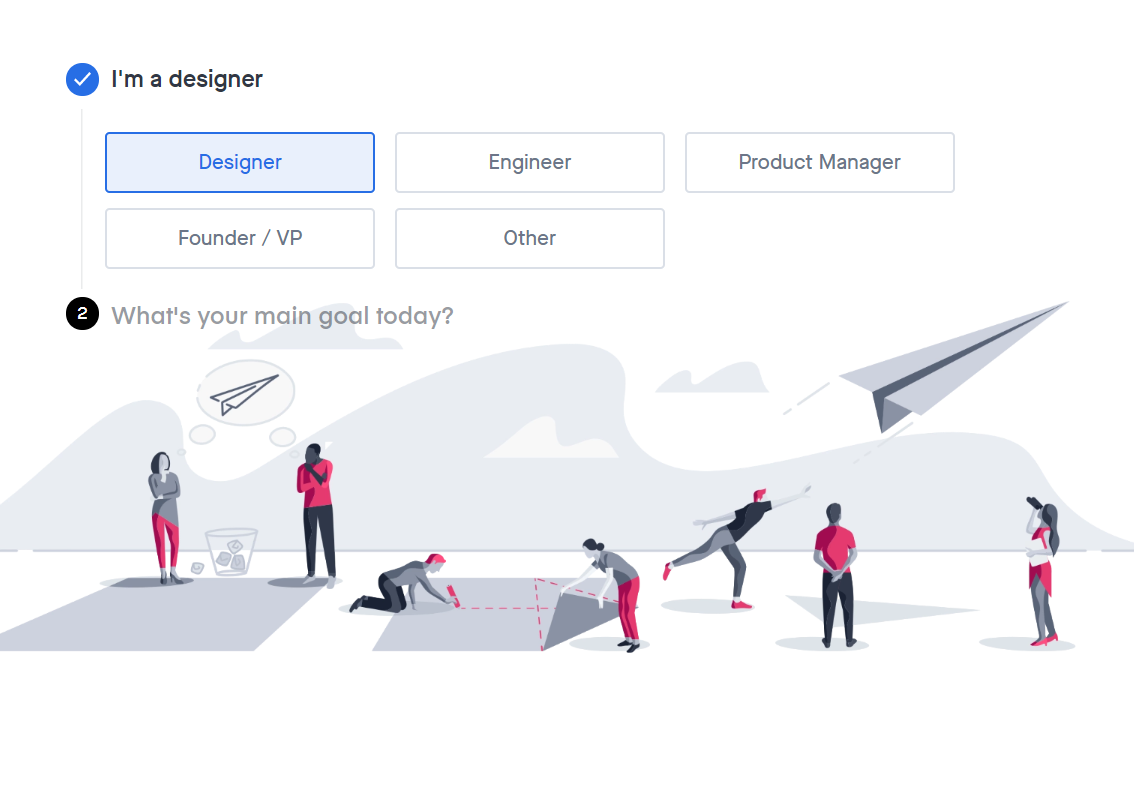
На начальном уровне пользователю предлагается выбрать роль и функции, которые он будет выполнять в данном проекте, как это представлено на рисунке 1.1.

Рисунок 1.1 - "Выбор роли в проекте"

Основными характеристиками InVision является:

* Возможность работы нескольких разработчиков над одним проектом.
* Все данные хранятся в облаке, что позволяет иметь доступ к проекту из любого места.
* Имеется база вопросов и задач с возможностью уведомления всей команды или отдельных разработчиков.

Система содержит в себе пакет инструментов быстрой обработки, что позволяет загружать в учетную запись пользователя графические материалы или новые прототипы и работать с ними прямо в браузере.

Инструменты и интерфейсы InVision являются очень простыми в работе и интуитивно понятными, что позволяет разработчикам с первых действий понять функции платформы и не тратить время на освоение интерфейса.

Благодаря кроссплатформенности системы можно работать не только на Windows, но и на Mac. Данная особенность является достаточно удобной, если проект синхронизирован, и работа над ним ведется с разных компьютеров. Это позволяет получать уведомления о новых файлах в проекте или комментариях в независимости от вашей операционной системы.

В случае работы всего над одним проектом, разработчикам предоставляется возможность бесплатного пользования. В остальных случаях, если вы собираетесь работать в команде или создавать несколько проектов, придется оплатить использование сервиса.

Рассмотрев основные возможности данной системы, можно сказать, что она является очень современной и динамичной системой для командной работы над созданием веб-проектов. Возможность работы с любой операционной системой и гибкого доступа к инструментам позволяет использовать эту платформу как сильный онлайн-инструмент для UX-разработки и тестирования.

## Фреймворк для веб-сайтов Bootstrap 4

Практически любая веб-страница содержит множество похожих компонентов, которые встречаются и на других сайтах. Это меню, навигация, элементы форм, заголовки и др. Не говоря уже про многоколоночную вёрстку, без которой вообще сложно превратить картинку макета в готовую веб-страницу. Чтобы меньше писать кода и по максимуму задействовать уже готовые решения применяются специализированные библиотеки, их часто называют фреймворками. Одним из таких фреймворков для вёрстки является Bootstrap.

Bootstrap разработали Марк Отто и Якоб Торнтон, сотрудники Twitter, именно поэтому в названии фигурирует имя компании. Их цель понятна любому разработчику — создать единый стандартный набор инструментов для сотрудников компании, ускоряющий их работу.

Во-первых, Bootstrap является самым популярным фреймворком, у его ближайшего конкурента в 3-5 раз меньше сообщество. Во-вторых, это не только css, но и js-фреймворк. То есть в Bootstrap написаны готовые стили и скрипты, для применения которых вам достаточно всего лишь прописать необходимые стилевые классы и атрибуты html-элементам.

На сегодняшний день Bootstrap давно перерос рамки одной компании, это открытый продукт, применяемый веб-разработчиками для вёрстки сайтов во всём мире.

Какие плюсы несёт в себе использование библиотеки.

1) Высокая скорость разработки

Фактически Bootstrap представляет собой конструктор, фрагменты которого вы включаете в свой проект при необходимости. Это уменьшает время разработки, потому что не требуется придумывать и писать их самостоятельно.

2) Адаптивный дизайн

Bootstrap направлен на создание макета под разные устройства — ноутбуки, планшеты, смартфоны. При этом код пишется один, а масштабирование в зависимости от ширины устройства берёт на себя фреймворк.

3) Открытая программа

Вы можете свободно изучать исходные коды библиотеки, изменять и расширять её под собственные нужды.

4) Совместимость с браузерами

Компоненты библиотеки написаны и протестированы с учётом работы разных браузеров. Это гарантирует, что макет будет выглядеть одинаково независимо от выбранного браузера.

4) Низкий порог вхождения

Чтобы использовать библиотеку в своей работе, требуется обладать минимальными знаниями по HTML, CSS и JavaScript. Это позволяет создавать эффектные сайты даже начинающим разработчикам.

5) Единая работа компонент

Bootstrap не просто вставляет какие-то элементы на страницу, но сразу же устанавливает их оформление и взаимодействие с пользователем через JavaScript. Вы получаете полностью работающий набор компонент, который достаточно добавить и настроить под себя.

Недостатки, конечно же, тоже имеются и они вытекают в основном из универсальности системы.

Файлы библиотеки, даже сжатые, занимают довольно много места и увеличивают нагрузку на сервер. С повышением опыта нужно загружать и устанавливать только требуемые компоненты Bootstrap, сокращая тем самым объём файлов.

Предлагаемые в библиотеке стили могут не подходить под дизайн разрабатываемого сайта и придётся много переделывать. В таких случаях, как известно, проще написать всё самому с нуля.

Так же минусом является, что кода обычно в библиотеке написано больше, чем если бы вы написали при разработке с нуля. Потому что, когда вы делаете самостоятельно, вы реализуете только необходимый функционал и все. В Bootstrap же есть все на все случаи жизни. Даже то, что вам может не пригодиться. Но опять же, эта проблема очень легко решается тем, что вы можете сами выбирать, какие компоненты фреймворка загрузить в css-файл. Например, вы вообще можете скачать только сетку, а все остальное делать самостоятельно.

Второй также минус – шаблонный дизайн. Да, действительно, я часто захожу на разные сайты и вижу там одинаковые кнопки. И я знаю, что они сделаны в Bootstrap, потому что уж слишком это очевидно. Но и эта проблема легко решается, потому что она будет существовать только в том случае, если вы будете использовать только готовые компоненты фреймворка и ничего никогда не кастомизировать под себя.

А если вы, например, подключите только сетку Bootstrap, то сможете воссоздать любой дизайн, при этом пользуясь очень удобной гибкой сеткой фреймворка.

Компоненты фреймворка

Bootstrap является всеобъемлющим фреймворком. Это означает, что в него заложено много компонентов. По сути, все, что может понадобиться при разработке типовых сайтов.

Это, например, выпадающее меню, кнопки, алерты, табы, индикаторы состояния, хлебные крошки, списки, заголовки и т.д. Если вы писали код для каких-то из этих компонентов, то наверняка знаете, что это делается не за 1 минуту. В Bootstrap же достаточно изучить немного сам фреймворк и все эти вещи вы сможете использовать очень быстро.

# 3. Spring Fraemwork

Spring Framework – универсальный фреймворк с открытым исходным кодом для Java-платформы. Spring может быть рассмотрен как коллекция фреймворков. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений, например, Inversion of Control-контейнер и фреймворк MVC.

Центральной частью Spring является контейнер Inversion of Control, который предоставляет средства конфигурирования и управления объектами Java с помощью рефлексии.

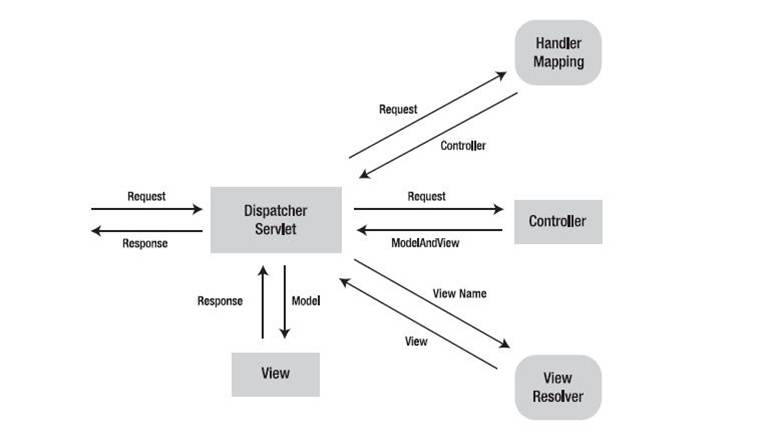
Контейнер отвечает за управление жизненным циклом объекта: создание объектов, вызов методов инициализации и конфигурирование объектов путём связывания их между собой. Объекты, создаваемые контейнером, также называются управляемыми объектами (beans). Обычно конфигурирование контейнера осуществляется путём загрузки XML-файлов, содержащих определение bean’ов и предоставляющих информацию, необходимую для создания bean’ов [1].

Spring MVC является фреймворком, ориентированным на запросы. В нем определены стратегические интерфейсы для всех функций современной запросно-ориентированной системы. Цель каждого интерфейса – быть простым и ясным, чтобы пользователям было легко его заново имплементировать, если они того пожелают. MVC прокладывает путь к более чистому front-end-коду.

## Spring MVC

Фреймворк Spring MVC обеспечивает архитектуру паттерна Model – View – Controller (Модель – Отображение – Контроллер) при помощи слабо связанных готовых компонентов. Паттерн MVC разделяет аспекты приложения (логику ввода, бизнес-логику и логику UI), обеспечивая при этом свободную связь между ними [3].

Model объединяет данные приложения. View отвечает за отображение данных, – как правило, генерируя HTML. Controller обрабатывает запрос пользователя, создаёт соответствующую Model и передаёт её для отображения во View.

Spring MVC построен вокруг центрального сервлета DispatcherServlet, который распределяет запросы по контроллерам, а также предоставляет другие широкие возможности при разработке веб приложений. Сервлет является интерфейсом Java, реализация которого расширяет функциональные возможности сервера. Сервлет взаимодействует с клиентами посредством принципа запрос-ответ. Процесс обработки запроса DispatcherServlet'ом показан на рисунке 1.1.

*Рисунок 1.1 – «Процесс обработки запроса DispatcherServlet'ом»*

Вначале DispatcherServlet получает запрос, далее он смотрит свои настройки, чтобы понять какой контроллер использовать. На данной схеме – это Handler Mapping. После получения имени контроллера запрос передается в Controller. В контроллере происходит обработка запроса и обратно посылается ModelAndView. DispatcherServlet на основании полученного ModelAndView ищет какое представление ему использовать (View Resolver) и получает в ответе имя представления View. В представление передаются данные (model) и обратно, если необходимо, посылается ответ от представления.