

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра Математического и компьютерного моделирования

**РАЗРАБОТКА WEB-ИНТЕРФЕЙСА ДЛЯ ИНТЕРНЕТ-
МАГАЗИНА С ИСПОЛЬЗОВАНИЕМ ФУНКЦИОНАЛЬНОГО
ЯЗЫКА ПРОГРАММИРОВАНИЯ HASKELL**

Автореферат бакалаврской работы

студентки 5 курса 561 группы
направления 09.03.03_ Прикладная информатика

механико-математического факультета

Лю Светланы Вадимовны

Научный руководитель
профессор, д.э.н, профессор

Л. В. Кальянов

Зав. кафедрой
зав. кафедрой, д.ф. - м.н.

Ю. А. Блинков

Саратов 2017

ВВЕДЕНИЕ

Актуальность выпускной квалификационной работы состоит в том, что доступность интернет-магазина, для любой категории предпринимателей, выступает как одно из главных преимуществ. При отсутствии свободного времени или знаний по управлению, обладая достаточным бюджетом можно купить готовую торговую площадку и в дальнейшем заниматься её продвижением.

Данная работа посвящена разработке модели данных для интернет-магазина одежды. Работа состоит из введения, трёх разделов, заключения, и приложения.

Целью работы является построение диаграммы прецедентов для выявления потребностей пользователей, кто будет пользоваться и что будет получать, а также создать Web-интерфейс и описать запросы.

Для создания информационной системы для выбранной предметной области перед нами стоят следующие задачи

- анализ предметной области;
- выделение сущностей и атрибутов базы данных;
- установление связей между сущностями;
- проектирование модели базы данных;
- создание Web-интерфейса.

При работе над выпускной квалификационной работой использовались следующие программы:

- SQLite;
- GHC;
- Web-Framework Yesod;
- Диаграммы PlantUML.

Данная работа состоит из трёх разделов:

- Обзор технологий использованных в работе
- Разработка информационной системы
- Создание интернет-интерфейса для магазина

Основное содержание работы

Первый раздел включает в себя краткий обзор Haskell, UML и Web-Framework Yesod, основных понятий БД, SQL, SQLite, PlanUML.

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем. UML не является языком программирования, но в средствах выполнения UML-моделей как интерпретируемого кода возможна кодогенерация.

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций, получивших название диаграмм. В терминах языка UML определены несколько видов диаграмм, каждая из этих диаграмм детализирует и конкретизирует различные представления о модели сложной системы в терминах языка UML, в данной работе была использована диаграмма прецедентов.

База данных — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ) [7].

SQL - формальный не процедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД). SQL основывается на исчислении кортежей.

SQL является прежде всего информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных. SQL можно назвать языком программирования, при этом он не является тьюринг-полным, но вместе с тем стандарт языка специ-

фикацией SQL/PSM предусматривает возможность его процедурных расширений.

Преимущества:

- Независимость от конкретной СУБД
- Наличие стандартов
- Декларативность

Недостатки:

- Несоответствие реляционной модели данных
- Сложность
- Отступления от стандартов
- Сложность работы с иерархическими структурами

Sqlite представляет из себя реляционную базу данных, используемую в качестве библиотеки при создании различных приложений. В отличие от многих других баз данных, таких как — MySQL, MS SQL Server, Oracle Database, ЛИНТЕР, Sqlite не является отдельным приложением, работающим по принципу «клиент-сервер», вместо этого база данных представляет из себя как бы библиотеку и работает в том же процессе, что и само приложение [9].

Язык Haskell – это чисто функциональный язык программирования. В императивных языках результат достигается при передаче компьютеру последовательности команд, которые он затем выполняет. При этом компьютер может изменять своё состояние. Например, устанавливаем переменную `a` равной `5`, производим какое-либо действие, а затем меняем её значение. Кроме того, у нас есть управляющие инструкции, позволяющие повторять несколько раз определённые действия, такие как циклы `for` и `while` [11].

Haskell – ленивый язык. Это означает, что он не будет выполнять функции и производить вычисления, пока это действительно не потребуется для вывода результата (если иное не указано явно). Подобное поведение возможно как раз благодаря ссылочной прозрачности. Haskell, будучи ленивым языком, пользуется этой возможностью и откладывает вычисления на то время, на какое возможно. Как только понадобится отобразить результаты, Haskell проделает минимум вычислений, достаточных для их отображения. Ленивость также позволяет создавать бесконечные структуры данных, потому что

реально вычислять требуется только ту часть структуры данных, которую необходимо отобразить.

Для создания сайта, выбираем Yesod, так как этот фреймворк пытается упростить процесс веб-разработки, играя на сильных сторонах языка программирования Haskell. Строгие проверки на этапе компиляции в языке Haskell затрагивают не только типы, а прозрачность по ссылкам гарантирует отсутствие непреднамеренных побочных эффектов. Сопоставление с образцом алгебраических типов данных позволяет учитывать все возможные случаи. Программируя на Haskell, избавляемся от целых классов ошибок [12].

Используя высокоуровневые декларативные приёмы, можно точно указать ожидаемые типы входных параметров. Да и сама обработка данных работает не так, как обычно – используя типобезопасные URL, можно быть уверенным в том, что выходные данные также сформированы правильно.

Веб-приложения содержат массу шаблонного кода. Где это возможно, Yesod старается использовать особенности языка Haskell, чтобы уменьшить количество работы:

- Библиотека для работы с формами в большинстве случаев уменьшает количество кода путём использования класса типов `Applicative`.
- Объявления маршрутов очень немногословны, но не в убыток безопасности типов.
- Код для сериализации и десериализации ваших данных генерируется автоматически.

В Yesod есть два вида кодогенерации. Для начала нового проекта предоставляется утилита, предназначенная для создания структуры файлов и каталогов. Тем не менее, большая часть кода генерируется на этапе компиляции благодаря механизму метапрограммирования. Таким образом, сгенерированный код никогда не устаревает. Простое обновление библиотеки автоматически обновит весь сгенерированный код.

ГHC, наиболее популярный компилятор языка Haskell, обладает потрясающими характеристиками производительности, и они постоянно улучшаются. Один только выбор языка программирования дал Yesod огромное преимущество в плане производительности по сравнению с альтернативами. Но

одного языка не достаточно — нам нужна архитектура, ориентированная на производительность.

Для примера рассмотрим подход к работе с шаблонами. Разрешив анализ HTML, CSS и JavaScript кода на этапе компиляции, Yesod не только избежал дорогостоящих операций ввода/вывода с диска, но и получил возможность оптимизировать рендеринг этого кода. Однако на этом архитектурные решения не заканчиваются. Благодаря тому, что лежащие в основе Yesod библиотеки используют такие продвинутые приёмы, как каналы (conduits) и строители (builders), можно быть уверенным в том, что код потребляет постоянное количество оперативной памяти, не приводя к исчерпанию файловых дескрипторов и других дорогостоящих ресурсов. Используя высокоуровневые абстракции, вы можете получить сильно сжатый и должным образом кешируемый код на CSS и JavaScript [15].

Yesod использует Шекспировское семейство шаблонных языков как стандартный подход к созданию HTML, CSS и Javascript. Это семейство языков имеет похожий синтаксис и общие принципы:

- Как можно меньшее вмешательство в языки, на базе которых создаются Шекспировские шаблоны, но в то же время использование преимуществ этих языков.
- Гарантии корректности содержимого обеспечиваются компилятором.
- Безопасность статической типизации, которая также предотвращает XSS (cross-site scripting) атаки.
- Автоматическая проверка валидности URL, где это возможно, с помощью типобезопасных URL.

Для передачи данных от пользователя Web-страницы на сервер используются HTML-формы. Для работы с формами в PHP предусмотрен ряд специальных средств.

Одно из главнейших достоинств PHP — то, как он работает с формами HTML. Здесь основным является то, что каждый элемент формы автоматически становится доступным вашим программам на PHP. Вот пример формы HTML:

```
1 <form action="action.php" method="post">
2 <p>Ваше имя: <input type="text" name="name" /></p>
```

```
3 <p>Ваш возраст: <input type="text" name="age" /></p>
4 <p><input type="submit" /></p>
5 </form>
```

Во втором разделе описываются диаграммы прецедентов и разработка модели базы данных.

Целями построения диаграммы прецедентов являются определение границы и контекста моделируемой предметной области на ранних этапах проектирования; формирование общих требований к поведению проектируемой системы; разработка концептуальной модели системы для ее последующей детализации; подготовка документации для взаимодействия с заказчиками и пользователями системы.

Админ - называют администратора вычислительной сети или же администратора веб-сайта. Данная система предоставляет следующие функции:

- Предоставление технической поддержки
- Администрирование серверов баз данных
- Изменение статуса заказа
- Изменение каталога товаров
- Просмотр заказов

Покупатель - зарегистрированный пользователь интернет-магазина. Данная система предоставляет «покупателю» следующие функции:

- Выбор товара
- Оформление заказа
- Оплата заказа
- Просмотр информации о товаре

Гость - незарегистрированный пользователь. Данная система предоставляет «гостю» следующие функции:

- Просмотр товара
- Регистрация на сайте

Курьер - посыльный интернет-магазина для передачи товаров. Данная система предоставляет «курьеру» следующие функции:

- Доставка товара внутри города
- Отчетность о доставке

Разработанная модель базы данных в соответствии с рисунком 1 состоит из следующих сущностей:

- Тип товара (идентификатор типа товара, наименование, вид товара)
- Склад (идентификатор склада, наименование, количество)
- Цена (идентификатор цены, стоимость)
- Оплата (идентификатор оплаты, наименование)
- Статус заказа (идентификатор статуса заказа, наименование)
- Страна (идентификатор страны, наименование)
- Производитель (идентификатор производителя, наименование, страна производителя)
- Товары (идентификатор товара, наименование, стоимость, количество на складе, размер, тип товара, производитель, цвет, материал)
- Акции (идентификатор акции, наименование, описание, товар, скидка)
- Доставка (идентификатор доставки, наименование)
- Заказы (идентификатор заказа, товар, количество, дата заказа, оплата, дата отправки, доставка, дата доставки, статус заказа)
- Покупатели (идентификатор покупателя, ФИО, адрес, телефон, заказ, почта, скидка)

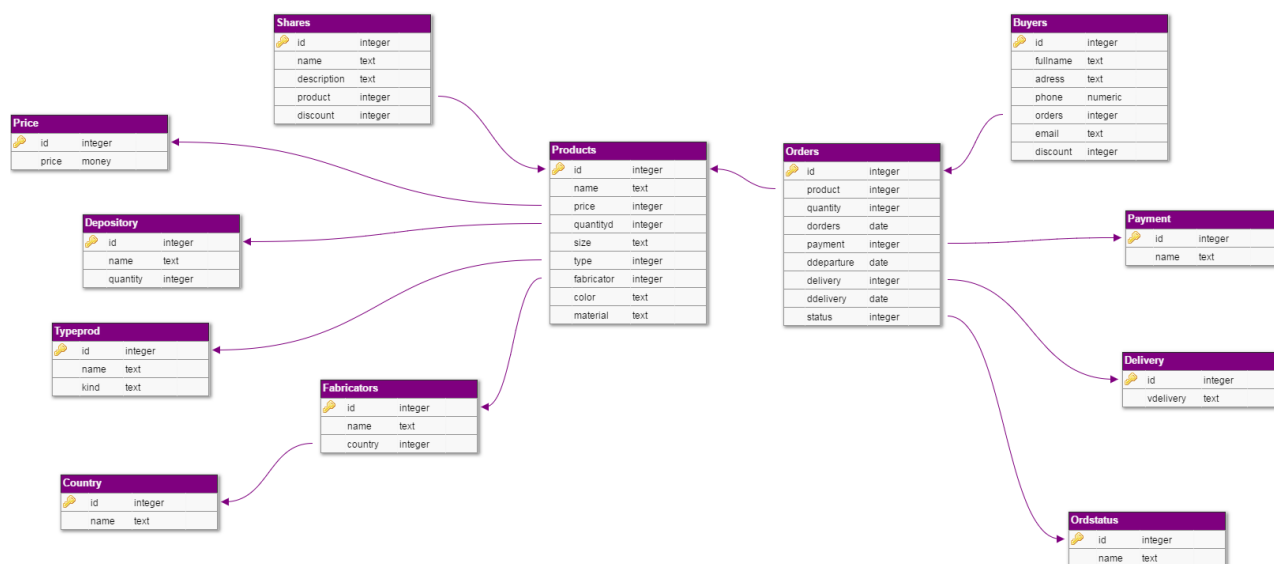


Рисунок 1 — Диаграмма логической модели базы данных

Третий раздел содержит в себе процесс создания интерфейса для интернет-магазина с помощью функционального языка программирования Haskell.

Для сборки Yesod используем `stack`. Порядок установки пакетов для Yesod Web Framework:

- Устанавливаем `stack`, следуя инструкциям¹
- Создаем новый каркас сайта: в командной строке вводим —
`stack new my-project yesod-sqlite && cd my-project`
- Далее устанавливаем необходимые пакеты:
`stack build yesod-bin cabal-install --install-ghc`
- Загружаем библиотеки: `stack build`
- Загружаем сервер: `stack exec -- yesod devel`
- Открываем Yesod сайт: «localhost:3000» и в соответствии с рисунком 2 видим:

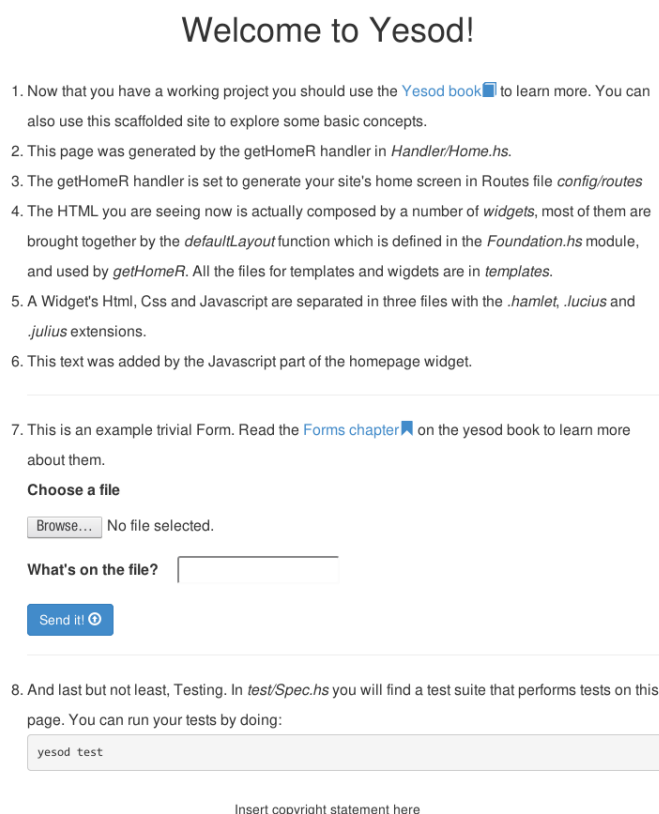


Рисунок 2 — Welcome to Yesod

¹The Haskell Tool Stack. [Электронный ресурс] - URL: <http://docs.haskellstack.org/en/stable/install/> (Дата обращения 09.03.2017)

Далее выбираем оформление в соответствии с рисунком 3 на главную страницу. После создаем виджеты «Women» и «Men» в соответствии с рисунком 4 для главной страницы, используя шаблоны. С помощью этих шаблонов, которые находятся в каталоге templates — Hamlet (HTML), Lucius (CSS), Julius (Javascript) редактируем интерфейс сайта и размещаем информацию, которая извлекается из БД с помощью файла «my-project.sqlite3».

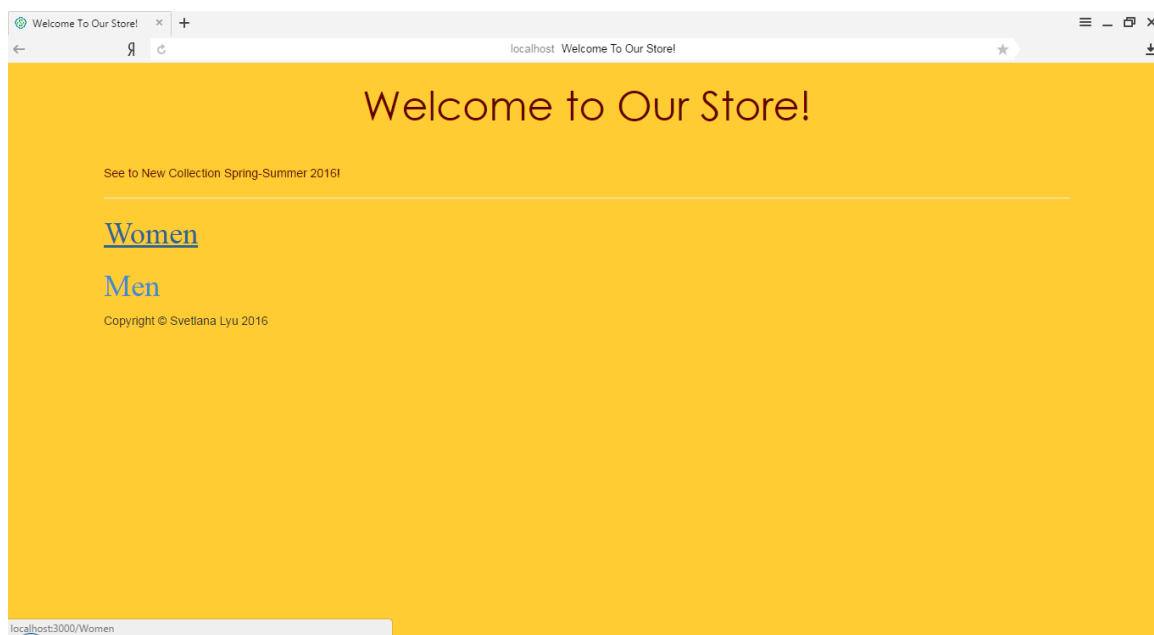


Рисунок 3 — Интерфейс готового сайта - Главная страница

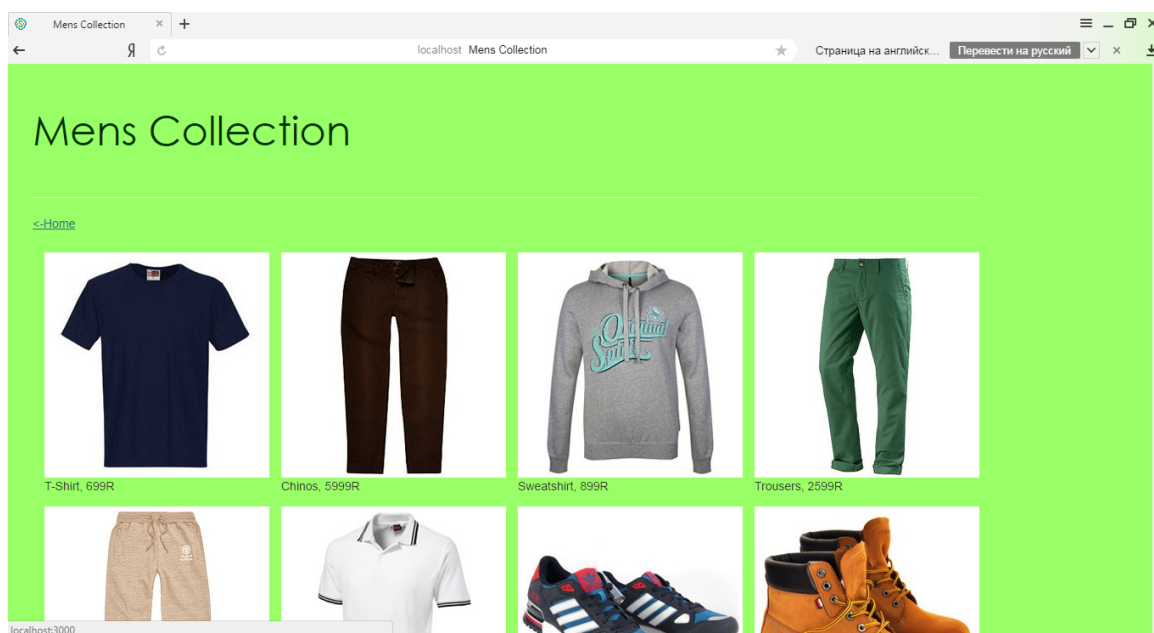
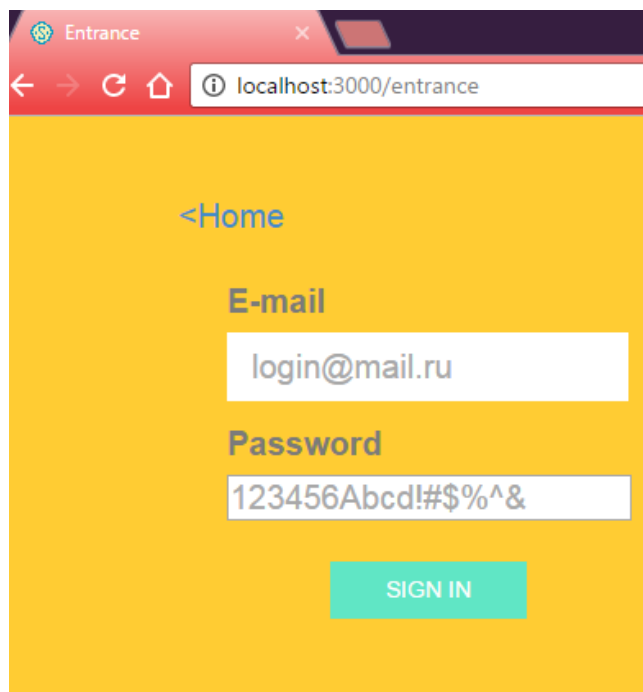


Рисунок 4 — Интерфейс готового сайта - «Men.hs»

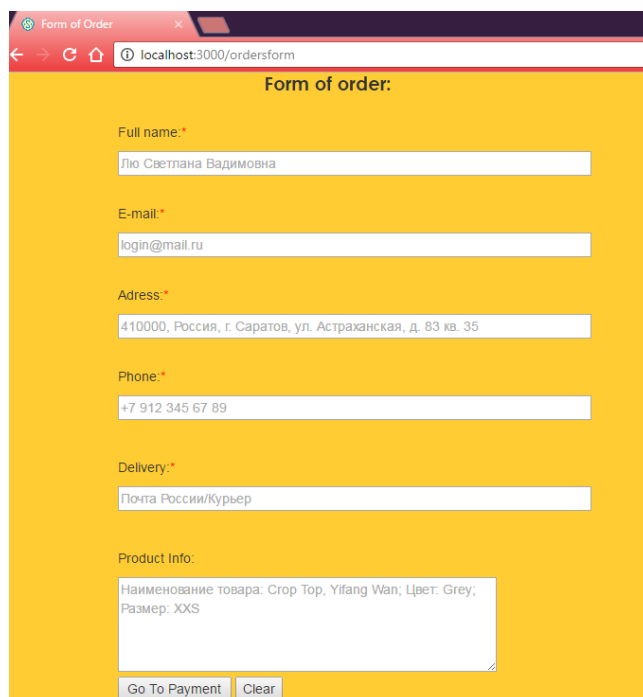
Так же созданы формы: входа, регистрации, оформления заказа и оплаты [20].

Форма входа в соответствии с рисунком 5 и форма оформления заказа в соответствии с рисунком 6:



The screenshot shows a web browser window with the title 'Entrance'. The address bar contains 'localhost:3000/entrance'. The page has a yellow background. At the top, there is a blue '<Home' link. Below it, the text 'E-mail' is displayed above a white input field containing 'login@mail.ru'. Underneath, the text 'Password' is displayed above another white input field containing '123456Abcd!#\$%^&'. At the bottom center, there is a green button with the text 'SIGN IN'.

Рисунок 5 — Вход на сайт



The screenshot shows a web browser window with the title 'Form of Order'. The address bar contains 'localhost:3000/ordersform'. The page has a yellow background. The title 'Form of order:' is centered at the top. Below it, there are several input fields: 'Full name:*' with 'Лю Светлана Вадимовна', 'E-mail:*' with 'login@mail.ru', 'Address:*' with '410000, Россия, г. Саратов, ул. Астраханская, д. 83 кв. 35', 'Phone:*' with '+7 912 345 67 89', and 'Delivery:*' with 'Почта России/Курьер'. Below these is a 'Product Info:' section with a text area containing 'Наименование товара: Crop Top, Yifang Wan, Цвет: Grey, Размер: XXS'. At the bottom, there are two buttons: 'Go To Payment' and 'Clear'.

Рисунок 6 — Оформление заказа

ЗАКЛЮЧЕНИЕ

В данной работе разработан интернет-интерфейс магазина одежды на языке программирования Haskell с помощью веб-фреймворка Yesod. Так же созданы формы входа, регистрации, оформления заказа и оплаты.

Для чего была построена и описана диаграмма прецедентов, и выявлены потребности пользователей. Построенная модель прецедентов есть во второй главе данной работы.

Коротко описан каждый запрос с прецедентами. Реализована постановка задач. На основе проведенного анализа была разработана БД. Так же обеспечена информационная потребность пользователей. Кратко описана разработка моделей функционирования системы, методология UML, а так же диаграмма вариантов использования. Проведен краткий обзор структурированного языка запросов (SQL), скриптового языка PHP, языка программирования Haskell и веб-фреймворка Yesod.