БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра технологий программирования

Лабораторная работа № 8
По курсу “Проектирование человеко-машинных интерфейсов”

Проектирование и разработка веб- и мобильного приложения

Методические указания по выполнению лабораторной работы

Подготовила: Давидовская М.И.,
Старший преподаватель кафедры ТП

Минск, 2022 г.

Содержание

Введение

В результате выполнения лабораторных работ студент должен приобрести следующие навыки и умения:

- проектирование ПО на основе человеко-центрированного подхода;
- разработка прототипов интерфейсов,
- применение принципов и шаблоны проектирования взаимодействия;
- анализ и оценка пользовательского интерфейса ПО;
- проектирование и разработка мобильного приложения (HTML5, CSS3) и веб-приложения (PHP/Python (Django)/Ruby (Ruby on Rails)/Java(Java Spring Framework: Spark, Spring MVC)/C#(Asp.Net MVC)); css-фреймворки (Bootstrap, Zurb Foundation), js- фреймворки (jQuery, AngularJS, React, Knockout, Vue) и др.

Лабораторные работы связаны между собой единой темой из предложенного преподавателем списка или выбранной студентом самостоятельно при условии предварительного согласования с преподавателем. Входными данными для каждой последующей работы являются результаты предыдущей.

Рекомендации по выполнению лабораторной работы

Цель работы — получить навыки проектирования и разработки мобильного приложения и веб-приложения с адаптивной верстской ((PHP/Python (Django)/Ruby (Ruby on Rails)/Java(Java Spring Framework: Spark, Spring MVC)/C#(Asp.Net MVC)); css-фреймворки (Bootstrap, Zurb Foundation), js- фреймворки (jQuery, AngularJS, React, Knockout, Vue) и др.

Выполнение лабораторной работы №8 состоит из следующих этапов:

2. Подготовить финальный отчет на основе технического задания (**см. требования в лабораторной работе 6**), который включает:
    - постановку задачи,
    - стратегию дизайна,
    - диаграммы бизнес-процессов и диаграммы вариантов использования,
    - диаграммы деятельности,
    - диаграммы классов и объектов,
    - диаграммы компонентов,
    - диаграммы развертывания,
    - схему базы данных, используя диаграммы "сущность-связь" (Entity RelationShip Diagram — ERD) и, при необходимости, диаграмму объектно-реляционного отображения (Object-relational mapping – ORD).
2. Завершить реализацию проекта согласно техническому заданию, включая хранение данных в базе данных, работу с сессиями, аутентификацию, валидацию данных и др.
3. Формировать автоматическую сборку при публикации изменений в репозиторий, используя сервис непрерывной интеграции (Сервис Github Actions).
4. Перед сборкой выполнять проверку качества кода с помощью сервиса Better Code Hub или иной, автоматические тесты и формировать сборку только после их прохождения.
5. Выполнить тестирование, внесение изменений и повторное тестирование. Типы тестов для веб-приложения должны соответствовать чек-листу https://habr.com/ru/post/542422/, а для мобильного — https://habr.com/ru/post/534190/.
6. Разработка иконки для мобильного приложения (если не подготовлена на предыдущих этапах) и логотипа и favicon.ico (для веб-приложения).
7. Разработать презентацию проекта, описывающую работы согласно ТЗ и вклад каждого участника проекта, включая информацию по комитам из git-репозитория и распределение работ в рамках проекта.
8. Для публикации на внешнем хостинге можно использовать бесплатные облачные платформы Render, OpenShisft, Heroku или Google Cloud. Для развертывания окружения разработки локально использовать контейнеры на примере Docker или LXC/LXD. Продемонстрировать подключение к окружению разработки и запуск приложения на облачном сервисе.

План выполнения лабораторной работы:

№	Задача
1.	Обновление технического задания.
2.	Завершить распределение задач на реализацию функционала и назначение ответственных.
3.	Разработать логотип, favicon.ico для веб-приложения и иконку для мобильного приложения.
4.	Завершить реализацию задач на разработку фронтенда и бекенда.
5.	Выполнить тестирование веб и мобильного приложения.
6.	Подготовить отчет и презентацию.

Рекомендации по использованию инструментов разработки

Что должен уметь frontend-разработчик

https://habrahabr.ru/company/netologyru/blog/327294/

Что должен уметь backend-разработчик

https://habrahabr.ru/company/netologyru/blog/328426/

https://geekbrains.ru/posts/profession_web_developer

http://bifot.ru/chto-dolzhen-umet-back-end-razrabotchik-developer-php/

https://kurspc.com.ua/node/389

https://jetbrains.ru/careers/requirements/backender/

70 полезных инструментов для фронтенд-разработчика https://proglib.io/p/frontend-workflow-instruments/

Непрерывная интеграциия, настройка, юнит-тесты и другие виды тестов

Автоматизация сборки

https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8

http://habrahabr.ru/post/155201/

Непрерывная интеграция и функциональное тестирование: два ключевых фактора разработки качественной информационной системы - http://www.epam.by/aboutus/news-and-events/articles/2011/aboutus-ar-07-14-2011.html

QA Automation - http://andrebrov.net/blog/qa-automation-часть-вводная/

QA Automation, часть 1: CI сервер наше все - http://andrebrov.net/blog/qa-automation-часть-1-ci-сервер-наше-все/

QA Automation, часть 2: Автотестирование – Начало. - http://andrebrov.net/blog/qa-automation-часть-2-автотестирование-начало/

Верстка и запросы

Генератор данных

https://www.mockaroo.com/

Perfect Pixel — верстка по макету

http://www.welldonecode.com/perfectpixel/

Как использовать Pixel Perfect — http://gearmobile.github.io/web-development/pixel-perfect/

Postman

Клиент для тестирования запросов к сайтам - https://www.getpostman.com/apps

Кроме десктопных приложений есть расширение для браузера Chrome. Для Firefox можно использовать другие расширения, например HttPrequest

Использование postman для тестирования запросов -

https://mindbox.fogbugz.com/default.asp?W1299

Настройка окружения

Vagrant

https://khashtamov.com/2015/12/vagrant-how-to-setup/

http://eax.me/vagrant/

Плаги Emmet https://babosik.ru/197-emmet-sublime-text-3-package-control.html
10 WebStorm Shortcuts You Need to Know
https://blog.jetbrains.com/webstorm/2015/06/10-webstorm-shortcuts-you-need-to-know/
Emmet With WebStorm, you can edit HTML and CSS code faster by applying Emmet features. Just type an Emmet abbreviation in HTML and press Tab to expand it into the markup. Emmet also works in the CSS and JSX context.
https://www.jetbrains.com/help/webstorm/emmet.html

Puppet

Кроссплатформенный клиент-серверное приложение, позволяющее централизованно управлять конфигурацией программ и операционных систем (как UNIX-подобных, так и семейства Microsoft Windows), установленных на нескольких компьютерах.

https://ru.wikibooks.org/wiki/Puppet

Docker

Принцип работы контейнеров можно легко понять использовав концепт виртуальных машин. Подобно виртуальным машинам, контейнеры обеспечивают безопасность путем одновременного запуска отдельных экземпляров операционных систем без взаимодействия друг с другом. Кроме того, как и виртуальные машины, контейнеры повышают мобильность и гибкость ваших проектов, так как вы не зависимы от какого-либо конкретного оборудования и можете перейти на любое другое облако, локальную среду и т.д.

Но в отличие от виртуальных машин, для которых требуется установка полнофункциональных операционных систем, что вызывает дополнительную

нагрузку. Контейнеры совместно используют ядро одной операционной системы, сохраняя при этом изоляцию по отношению к другим контейнерам. Проще говоря, вы получаете тот же результат, что и при использовании виртуальных машин, но без дополнительной нагрузки.

Docker использует такую же схему для создания контейнеров на VM на базе Linux. В одном контейнере Docker, вы получаете доступ ко всем необходимым вам ресурсам: исходному коду, зависимостям и среде выполнения.

https://www.hostinger.ru/rukovodstva/kak-ustanovit-wordpress-na-docker#--Docker

https://blog.amartynov.ru/docker-%D0%BD%D0%B0-windows-%D1%83%D0%B6%D0%B5-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D0%B5%D1%82/

https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-docker/configure-docker-daemon

Препроцессоры

https://zyubin.ru/frameworks/bootstrap/bootstrap-less-rabota-s-preprocessorom.html

https://htmlacademy.ru/courses/85

http://paulradzkov.com/2017/local_variables/

https://habrahabr.ru/post/214143/

Краткая инструкция по экспорту в css

1. Создайте файл style.less, в котором вы импортируете необходимые бутстрап-файлы (пусть они лежат в отдельной папке, чтобы не путаться). Например:

```
@import "bootstrap/less/variables.less"
@import "bootstrap/less/mixins.less"
@import "bootstrap/less/normalize.less"
@import "bootstrap/less/mixins.less"
@import "bootstrap/less/buttons.less"
```

Список всех файлов вы можете увидеть в bootstrap.less. Если вам нужен весь бутстрап, вы можете просто этот bootstrap.less импортировать.

2. Пишите свои стили после импортов или в отдельном файле, и включите его в style.less с помощью импорта:

```
// Bootstrap
@import "bootstrap/less/variables.less"
...
@import "bootstrap/less/buttons.less"

// Project Styles
@import "project.less"
```

3. Скомпилируйте LESS в CSS с помощью, как уже посоветовали, npm-пакета less или утилиты Prepros. Подключайте итоговый CSS-файл к вашей странице.

npm install -g less
lessc styles.less > styles.css

Frontend frameworks

Как выбрать фреймворк для frontend-разработки https://habrahabr.ru/post/277547/

Обзор 5 самых популярных JavaScript фреймворков и библиотек 2017 https://habrahabr.ru/post/321844/

5 самых популярных фреймворков для JavaScript https://geekbrains.ru/posts/5_js_frameworks

VUE — https://habrahabr.ru/post/329452/

Автоматизация frontend-разработки (taskrunners)

Автоматизация для фронтендеров https://www.youtube.com/watch?v=rJr0-e0ZJiU

https://habrahabr.ru/post/251807/

Level Up для новичков: gulp и requirejs https://habrahabr.ru/post/264869/

https://habrahabr.ru/company/2gis/blog/269743/

Пособие по webpack https://habrahabr.ru/post/309306/

React с нуля. Настройка связки gulp + webpack + babel + react https://medium.com/@artvaleyev/react-s-nulya-nastrojka-svyazki-gulp-webpack-babel-react-7be0203614bc

https://loftblog.ru/lessons/avtomatizaciya/

Сервисы временной почты

https://temp-mail.org/

https://temp-mail.ru/

https://dropmail.me/ru/