

2020. ИТИС. ИНФОРМАТИКА.

СЕМЕСТРОВАЯ РАБОТА 1 (ВЕБ-ПРОЕКТ)

11-905. 2 КУРС 3 СЕМЕСТР

1. Требования к реализации проекта

Технологический стек: Web-приложение на Java Servlets, сервер Tomcat, шаблонизатор FreeMarker (но можно обсудить JSP+JSTL), БД postgresql или mysql, html, css (с библиотекой bootstrap или альтернативной), js с библиотекой jquery

Правильная организация работы: наличие задач в трекере, хорошая работа с коммитами в репозитории – правильные коммит-сообщения, соблюдение дедлайнов, скринкаст, презентация.

Концепции, модели, паттерны: MVC/MVP, ORM (для обработки БД) со слоем сервисов и паттерном проектирования DAO или Repository.

Минимальный функционал:

- Авторизация, регистрация, запомнить меня
- Профиль пользователя
- Просмотр информации, внесение информации
- Разделение возможностей между анонимным пользователем и зарегистрированным пользователем.
- Сообщения между пользователями/форум/иные способы общения
- Поиск каких-либо сущностей на сайте с возможностью его настройки (*гугл. настройка поиска/фильтры*)

Требования к работе приложения:

- Использование сессий (для авторизации как минимум)
- Использование cookie (запомнить меня минимум)
- Использование include и наследования (ОБА СПОСОБА) в шаблонах
- Не менее 3 форм с наличием различных виджетов – text, textarea, select, check, file-field
- Написание небольшого js-функционала на сайте (подсчет количества символов с запретом ввода большего их числа, автозаполнение форм и т.п.), написанного верстальщиком с нуля. Валидация данных не относится к данному требованию.
- Использование 2х любых js-виджетов любой библиотеки (например, bootstrap) – модальные окна, слайдеры, выпадающие меню и т.п.
- Использование аjax-запросов для поиска/проверки (хотя бы 1)
- Валидация данных (на клиенте и сервере), вводимых пользователем (здесь не менее 2х регулярок).

- Не менее 5 пунктов меню
- Не менее 10 разных страниц на сайте.
- Не менее 5 сущностей данных с не менее 3мя полями (не считая primary key) – промежуточные таблицы для m-m не считаются.
- Хэширование паролей (гуглим MessageDigest)
- Уведомление пользователя об ошибках при вводе информации на сайт.

2. Роли с должностными обязанностями:

Dev1

Собирает роли серверного (back-end) разработчика, техлида, архитектора, менеджера проекта. Фактически, ответственный за проект (получает за него основные баллы).

- заводит task-трекер, раздает задачи, следит за их выполнением,
- заводит чат проекта
- заводит хранилище информации о проекте (wiki-движок, иное место, куда выкладывается вся документация, полезные сведения, ссылки по проекту и тп.)
- разрабатывает архитектуру приложения
- проектирует базу данных (сущности)
- заводит репозиторий
- пишет серверный код, отлаживает приложение
- прикручивает код верстальщика, превращая его в шаблоны
- осуществляет баг-фиксинг проекта
- следит за соблюдением релизов и дедлайнов,
- презентует проект
- Записывает итоговый скринкаст, описывающий проект

Dev2

Собирает роли аналитика, тестировщика, проектировщика интерфейса, дизайнера, верстальщика, front-end разработчика – ответственный за проработку предметной области проекта, за работоспособность проекта, за интерфейс, дизайн и клиентский код

- собирает требования заказчика
- готовит документ с требованиями (прочитать разницу в функциональных / пользовательских / бизнес требованиях)
- по вариантам использования проектирует интерфейс
- по интерфейсу проектирует дизайн (возможно делать его сразу вместе с версткой)
- верстает дизайн (на выходе – перечень html, css, js, который передается разработчику)
- исправляет ошибки верстки

- начиная с рабочей версии, тестирует проект, пишет отчеты по тестированию, ставя соответствующие задачи разработчику и верстальщику.
- ответственен за наполнение сайта контентом для презентации

Product Owner.

- собранные требования аналитик команды согласует с ним. Проект не должен превышать технологий и требований, описанных в начале файла
- Ответственен за приемку работ – должен принять проект в рамках описанных требований, не давая поблажек.

3. Сценарий работ

- Dev2 собирает требования у РО, составляет документ с требованиями.
- Dev1 заводит таск-трекер, заводит чат, заводит хранилище информации, проектирует базу данных.
- Dev2 по требованиям проектирует интерфейс, после согласования создает дизайн и делает верстку.
- Dev1 по требованиям, модели данных, интерфейсу создает архитектуру приложения (*здесь - перечень запросов, перечень сервлетов с их описанием*), заводит репозиторий, создает проект, начинает разработку.
- Dev1 следит за выполнением сроков, выпуском релизов, фиксирует задачи.
- Dev2 выполняет простое тестирование, пишет тест-отчеты, передает их Dev1 и Dev2 в виде задач.
- Dev1 в ветке master хранит рабочую версию вашего проекта.
- Dev1 и Dev2 исправляют баги, разрабатывают новый функционал, если это требуется.
- Dev2 наполняет сайт контентом для презентации.
- Dev1 записывает англоязычный скринкаст за несколько дней до презентации.
- Dev1 презентует проект на английском.
- РО принимает работу.

4. План-график работ

17 сен - 23 сен	Сбор требований, подготовка документации, проектирование интерфейса, БД, приложения. Согласование требований.
-----------------	---

24 сен – 30 сен	Начало разработки, Верстка
01 окт – 14 окт	Разработка, тестирование, баг-фиксинг
15 окт – 21 окт	Завершение разработки, финальное тестирование
22 окт – 28 окт	Подготовка презентации и скринкаста
02 - 04 ноября	Презентация

5. Разбалловка

Баллы Dev1

Работа	Балл
Наличие работы с задачами в таск-трекере	2
Наличие регулярных коммитов (не менее 7-10 существенных коммитов в неделю)	2
Авторизация, регистрация, запомнить меня, хеширование пароля	1 (0.25 за каждый)
Профиль пользователя	1
Создание/редактирование информации, доступной для просмотра посетителям	1
Загрузка пользователем файлов	1
Разделение возможностей между анонимным пользователем и зарегистрированным пользователем	1
Сообщения между пользователями/форум/иные способы общения	1
Поиск каких-либо сущностей на сайте с возможностью его настройки (<i>гугл. настройка поиска/фильтры</i>)	2 (1 за просто поиск)
Использование аjax-запросов для поиска/проверки (хотя бы 1)	1

Использование include и наследования (ОБА СПОСОБА) в шаблонах	1 (0.5 за каждое)
Валидация данных на сервере с помощью регулярных выражений (адекватная, не менее 2х)	1 (0.5 за каждую)
Не менее 5 сущностей данных с не менее 3мя полями (не считая primary key) – промежуточные таблицы для m-m не считаются.	1 (0.2 за каждую с 3мя полями)
Написана ORM для работы с БД	1
Реализован паттерн DAO или Repository	1
Подготовленный вовремя документ с моделью данных в БД	1
Подготовленная вовремя архитектура запросов	1
Адекватность и работоспособность итогового приложения	2
Оценка группы по итогам презентации	1
Оценка РО	1
МАКС. ИТОГ	24

Баллы за скринкаст и презентацию ставятся в курс английского языка для Dev1, если не будет оговорено обратного

Баллы Dev2

Не менее 3 форм с наличием различных виджетов – text, textarea, select, check, file-field	1 (0.3 за каждую форму, вычитается 0.1 за отсутствие виджета)
Написание небольшого js-функционала на сайте (подсчет количества символов с запретом ввода большего их числа, автозаполнение форм и т.п.) с нуля.	1
Валидация данных на клиенте с помощью регулярных выражений (адекватная, не менее 2х)	1
Использование 2х любых js-виджетов любой библиотеки (например, bootstrap) – модальные окна, слайдеры, выпадающие меню и т.п.	1
Не менее 5 пунктов меню	1

Не менее 10 разных страниц на сайте.	1
Подготовленный вовремя документ с требованиями	1
Подготовленный вовремя проект интерфейса	1
Своевременное предоставление верстки Dev1 (по оценке Dev1)	1
<i>МАКС. ИТОГ</i>	9

Таким образом, максимальный балл за семестровую: $24 + 9 = 33$