**Слайд 1**

Здравствуйте, меня зовут Мельник Сергий Романович я студент группы 22919/1. Сегодня я вам расскажу о своём проекте «Электронный Дневник Политеха»

*На слайде есть QR код ссылающийся на мой GitHub.*

**Слайд 2**

**Начнем с Предметной Области.**

Электронный дневник который должен работать на всех популярных браузерах (Google Chrome, Microsoft Edge, Яндекс), планируется до 35 тыс. пользователей одновременно *(на официальном сайте политеха я узнал сколько всего имеется студентов в Питерских филиалах, и добавил к ним количество преподавателей и системных администраторов)*, 4 уровня доступа, *(слайд 4)* создание backup-ов защита данных *(учетной записи, а так же данных самого журнала)* как при хранении, так и передаче стандартными протоколами защиты, *(ГОСТ Р 50922-2006, ГОСТ Р 51275-2006, ГОСТ Р 53114—2008)* на выполнение данного проекта предполагается 1 год времени.

**Слайд 3**

**Перейдём к Модели Жизненного Цикла.**

Мною была выбрана комбинация из двух Моделей Жизненного Цикла: Прототипирование и Каскадная модель. Смысл данного объединения в том, чтобы нивелировать минусы Каскадной модели плюсами Прототипирования.

**Каскадная**

Плюсы модели ЖЦ:

* У нас есть возможность вернуться на предыдущие этапы.
* Мы получим в конце готовый продукт.
* У нас есть чёткий план.

Минусы модели ЖЦ:

* У нас нет всех требований на начальном этапе.

**Прототипирование**

Плюсы модели ЖЦ:

* У нас обеспечиваются все требования к ПО, благодаря созданию прототипов.
* Мы можем в полном объёме определить требования или же их изменить в процессе прототипирования.

Пройдёмся по каждому этапу:

1. **Разработка прототипа + требований** (создаём на основе начальных требований прототип и показываем заказчику, и повторив такое действие несколько раз мы получим прототип и полный набор чётких требований).
2. **Планирование** (график выполнения проекта).
3. **Дизайн** (Создание красивого пользовательского интерфейса).
4. **Кодирование** (Написание кода html, css и js для нашего сайта).
5. **Тестирование** (Тестирование продукта на работоспособность удобность и т.д.).
6. **Инсталляция** (Проверка системных требований, хостинг сайта).
7. **Сопровождение** (Техническая поддержка проекта, фиксы баг, оптимизация).

**Слайд 4**

**Создание графического интерфейса.**

При разработке использовались 3 принципа удобного Графического интерфейса:

* Принцип простоты
* Принцип видимости
* Принцип повторного использования

В данном проекте реализованы 4 уровня доступа:

* Студент
* Преподаватель
* Администрация
* Сисадмин

**Слайд 5**

*На слайде отображена Навигационная карта Сайта.*

**Слайд 6**

На экране вы можете видеть ранжированный список функционала, который планируется в проекте. Основным функционалом является возможность пользователей просматривать, выставлять, изменять оценки; создавать расписание для студентов и добавлять домашнее задание.

**Слайд 7**

При создании руководства оператора использовалась Единая система программной документации.

В нашем руководстве оператора (обычный пользователь) отображены 2 случая, когда пользователь может увидеть поясняющие системные сообщения.

**Неверный логин или пароль**

Системное сообщение: “Не верный логин или пароль”

При вводе некорректных логина или пароля во время Аутентификации, поля для ввода данных обведутся красным и будет выведено сообщение о вводе некорректных логина или пароля.

Решение ошибки:

Попытаться ввести правильные Пароль и Логин.

**Забыли пароль?**

Системное сообщение: “Забыли пароль?”

При частом вводе некорректных логина или пароля во время Аутентификации, высвечивается предложение восстановить Логин или Пароль.

Решение ошибки:

Нажать на кнопку “Нет” если восстановление не требуется, нажать кнопку “Да” если восстановление требуется.

*На экране есть 2 примера ошибок и сообщений при их возникновении.*

**Слайд 8**

|  |  |  |
| --- | --- | --- |
| Критерий | Метрика | Оценочные критерии |
| Надёжность | Способность к восстановлению | **Резервное копирование данных каждую неделю.** |
| Зрелость | **Время работы без сбоев не менее 12 часов.** |
| Производительность | Временная эффективность | **Загрузка любой страницы сайта не более 2 секунд.** |
| Эффективность использования ресурсов | **Не нагружать сервера более чем на 80%** |

Тест кейсы создавались по регистрационной форме Электронного дневника.

Использовался функциональный вид тестирования, а также использовались 3 метода тестирования:

* Причина / Следствие (Cause/Effect — CE)
* Анализ Граничных Значений (Boundary Value Analysis — BVA)
* Метод эквивалентного разделения (Equivalence Partitioning — EP)

**Слайд 9**

Также была создана двумерная таблица, содержащая соответствие функциональных требований продукта и подготовленных тестовых сценариев - Traceability matrix.

Тестовое покрытие составило 36%

**Слайд 10**

Также в проекте реализовано удобное отслеживание процесса тестирования при помощи веб-инструмента для организации и управления процессами тестирования на всех уровнях – TestRail.

**Слайд 11**

*На слайде ссылка на макет регистрационной странички сайта (хост - мой комп)*

**Слайд 12**

**Вывод:**

**Я подсчитал, что моя работа выполнена на 33%, так как не все обещанные функции реализованы, в дальнейшем планируется продолжать работу над этим проектом, чтобы доделать обещанные заказчику функции и реализовывать новые при надобности.**

**В следующий раз нужно будет сделать по-другому: набрать более опытных разработчиков, увеличить финансирование проекта, .**

**Особенно мне удалось выполнить работу по тестированию продукта.**

СПАСИБО ЗА ВНИМАНИЕ. ГОТОВ ОТВЕТИТЬ НА ВАШИ ВОПРОСЫ.