**Курсов проект по дисциплината „Управление на софтуерни проекти“**

*План за изпълнение*

Изготвил: Ерика Карамучева

ФН: 2101321067

Специалност: Софтуерно инженерство

1. Потребителски истории и оценки

* Регистрация в системата
* Като клиент (студент, преподавател, университетска администрация) искам да мога да се регистрирам в приложението с университетски акаунт, за да имам достъп до всички функции.
* Точки: 8
* Критерии за приемане:
* Потребителят може успешно да се регистрира с валиден университетски акаунт.
* Системата валидира невалидни акаунти и връща съответна грешка.
* Регистрираният потребител получава потвърждение и има достъп до функциите.
* Публикуване на материали
* Като преподавател, аз искам да мога да публикувам материали за лекциите, за да ги споделям с моите студенти.
* Точки: 5
* Критерии за приемане:
* Преподавателят може да качва файлов в различни формати – pdf, ppt, docx, xlsx.
* Качените файлове могат да бъдат достъпени от студентите в съответния курс.
* Интерфейсът е лесен за използване и потвърждава успешното качване.
* Преглед на разписание
* Като студент, аз искам да мога да преглеждам разписанието си за седмицата, за да планирам времето си ефективно.
* Точки: 8
* Критерии за приемане:
* Потребителят може да вижда разписание за текущата и следващата седмица.
* Разписанието се зарежда динамично от университетската система.
* Промени в разписанието се отразяват автоматично и в реално време.
* Разписанието може да бъде филтрирано по ден или предмет.
* Добавяне на събития в календара
* Като преподавател, аз искам да добавям събития като консултации или изпити в календара, за да информирам своите студенти за предстоящите събития своевременно.
* Точки: 13
* Критерии за приемане:
* Преподавателят може да събитие, като зададе дата, час, място и описание.
* Събитията са видими за всички студенти, които са записани в съответния курс.
* Системата изпраща известие на студентите за новосъздадени събития.
* Преподавателят може да редактира или изтрива събития.
* Известия за промени в графика
* Като студент, аз искам да получавам известия за промени в графика, за да не пропускам лекции и събития.
* Точки: 8
* Критерии за приемане:
* Потребителят получава известия в реално време за промени в графика.
* Известията са налични в приложението и като push нотификация.
* Системата съхранява хронология на известията.
* Оценки и обратна връзка
* Като студент, аз искам да виждам оценките си за всички курсове, за да знам какъв е напредъка ми по всеки предмет.
* Точки: 8
* Критерии за приемане:
* Оценките са видими на страницата на курса и са свързани със съответните задания.
* Потребителят може да види подробности за всяка оценка, включително обратна връзка от преподавателя.
* Системата изпраща известие, когато бъде въведена нова оценка.
* Данните са видими само за студента, влезнал в системата.
* Управление на потребителите
* Като част от университетската администрация, аз искам да мога да управлявам потребителите и съдържанието в приложението, за да осигуря коректно използване.
* Точки: 13
* Критерии за приемане:
* Администраторът може да добавя, редактира и премахва потребители.
* Правилно се определя ролята на потребителя- студент, администратор, преподавател.
* Всички промени се записват в логове за сигурност.

1. Технически процес

* Спринт организация
* Продължителност на спринт: 2 седмици
* Капацитет на екипа: 30 точки за спринт
* Общ брой планирани спринтове:
* Срещи
* Ежедневна среща (Daily Scrum)
* Време: 10:30 всеки работен ден
* Продължителност: 15 минути
* Участници: целият екип
* Планиране на спринт
* Време: Първия понеделник на спринта
* Продължителност: 2.5 часа
* Участници: целият екип
* Преглед на спринт
* Време: Последния петък на спринта
* Продължителност: 2 часа
* Участници: екип и заинтересовани страни
* Ретроспективна среща
* Време: Последния петък на спринта (след срещата за преглед)
* Продължителност: 1 час
* Участници: Екип
* Definition of Done
* Кодът е прегледан от поне един друг разработчик
* Всички тестове преминават успешно
* Кодът е документиран
* Функционалността е тествана на различни устройства
* Изпълнени са всички критерии за приемане
* Документацията е актуализирана
* Тестовете за изпълнение показват приемливи резултати
* Технически практики
* Продължителна интеграция с автоматизирани тестове
* Code review процес за всички промени
* Автоматизиран deployment
* Мониторинг за качеството на кода
* Ежедневни бекъпи на кода и данните

1. Разпределение на спринтовете

* Спринт 1 (Седмица 1-2) – подготовка и планиране
* Разработване на техническа спецификация
* Създаване на потребителски истории
* Настройка на среди за разработка
* Изграждане на базова архитектура
* Спринт 2 (Седмица 3-4) – базови backend функционалности и интеграция с университетски системи
* Проектиране на базата данни
* Започване на интеграция с университетските системи (основна цел – за реализация на регистрацията)
* Разработка на API за регистрация и вход
* Дизайн на интерфейси ( начален екран, вход в системата)
* Спринт 3 (Седмица 5-6) - разработка backend функционалности и интеграция с университетски системи
* Завършване на интеграцията с университетските системи
* API за дигитални класни стаи и качване на файлове
* Създаване на интерфейс за визуализация на класни стаи и файлове
* Тестване на качването на файлове и администриране на класна стая
* Спринт 4 (Седмица 7-8) – управление на разписанията
* Backend API за разписания
* Frontend API за разглеждане и филтър на разписания
* Тестване на функционалността за разписания
* Спринт 5 (Седмица 9-10) – система за оценки
* Backend API за оценки
* Frontend API за разглеждане на оценките
* Тестване на функционалността за оценки
* Спринт 6 (Седмица 11-12) - нотификации
* Създаване на система за изпращане на нотификации
* Визуализация на нотификации
* Тестване на нотификациите
* Спринт 7 (Седмица 13-14) – разработка на чат функционалности
* Разработка на функционалностите за чат и групи
* Интеграция с функционалността за нотификации
* Разработване на дизайна за чат
* Тестване на функционалността за разговори
* Спринт 8 (Седмица 15-16) – сигурност и роли
* Имплементация на сигурността (шифроване на данни, защита на API с JWT)
* Разработване на система за роли и права
* Разработка на интерфейс за администриране на роли и права
* Тестване на системата за роли и права
* Спринт 9 (Седмица 17-18) – UX/ UI оптимизация
* Тестове за потребителско удобство
* Събиране на обратна връзка за интерфейса
* Подобрения в UX/ UI
* Спринт 10 (Седмица 19-20) – мобилна адаптация
* Специализирано тестване на платформата за мобилни устройства и оптимизация на UI/UX
* Добавяне на функционалността offline mode за разписанията
* Оптимизация на зареждането на ресурси
* Спринт 11 (Седмица 21-22) – потребителско тестване
* Подготовка на сценариите за тестване
* Провеждане на тестове
* Събиране на обратна връзка
* Спринт 12 (Седмица 23-24) – оптимизация на производителността
* Оптимизация на backend заявки
* Подобрения на frontend за по- бързо зареждане
* Спринт 13 (Седмица 25-26) – финални подобрения
* Финална оптимизация
* Документация
* Подготовка за deployment
* Спринт 14 (Седмица 27-28) – внедряване и корекции
* Deployment
* Мониторинг
* Финални корекции

1. Метрики за проследяване

* Velocity на екипа
* Процент завършени истории за спринт
* Време за отстраняване на бъгове
* Покритие на кода с тестове
* Потребителска удовлетвореност
* Производителност на приложението