**План разработки**

1. Составление технического задания
   1. Разработка и утверждение технического задания
      1. Определение требований к программе

Общие описание системы

Необходимо разработать систему дистанционного обучения, позволяющую преподавателям создавать и размещать электронные учебные курсы для использования в учебном процессе, в том числе для проведения рубежного контроля по отдельным темам и/или итоговой аттестации; студентам проходить созданные курсы в процессе изучения дисциплин, что обеспечивает более глубокую самостоятельную проработку материала изучаемой дисциплины.

Термины и определения

Гость - неавторизованный пользователь, который имеет возможность просматривать главную страницу системы и регистрироваться.

Требования к функционалу системы

Система должна поддерживать четыре типа пользователей: гость, преподаватель, студент, администратор. Система должна позволять регистрацию и авторизацию. Администратор должен иметь возможность управлять аккаунтами пользователей.

Функции пользователя типа преподаватель:

* Система должна позволять создать курс, задать его название и аннотацию.
* Один преподаватель может создать неограниченное число курсов.
* Система должна позволять в рамках курса создать темы для изучения, задать их название и описание.
* Система должна позволять привязать к созданной теме учебные материалы: видео-материал, презентация, текстовый документ.
* Система должна позволять установить ограничения на доступность материалов по дате и по оценке (для доступа к материалам студенту необходимо достичь установленной оценки).
* Система должна позволять оставлять комментарии под учебными материалами.
* Система должна позволять проводить вебинары в рамках курса.
* Система должна позволять установить ограничения по дате на доступность вебинара.
* Система должна позволять оставлять комментарии к вебинару.
* Система должна позволять создавать новости.
* Система должна позволять оповещать студентов о публикации новости.
* Система должна позволять принудительно подписывать студентов на оповещения о новостях.
* Система должна позволять создавать тесты.
* Система должна позволять ограничить время прохождения теста.
* Система должна позволять установить ограничения на доступность теста: 1) по дате; 2) по оценке (для доступа к материалам студенту необходимо достичь установленной оценки).
* Система должна позволять создавать задания с развернутым ответом.
* Система должна позволять ограничить время прохождения задания.

Функции пользователя типа студент:

* Система должна позволять записаться на курс.
* Система должна позволять оставлять комментарии под учебными материалами.
* Система должна позволять оставлять комментарии к вебинару.
* Система должна поддерживать средство коммуникации между студентами и преподавателями
  + 1. Определение стадий, этапов и сроков разработки [программы и документации](javascript:void(11448)) на нее

|  |  |
| --- | --- |
| **Название этапа разработки** | **Сроки** |
| Определение инструментов и методов, которые будут использованы при разработке | 24.09.2016 – 08.10.2016 |
| Определение состава документации | 24.09.2016 – 08.10.2016 |
| Подготовка и настройка необходимых инструментов | 08.10.2016 – 22.10.2016 |
| Написание первых версий плана конфигурационного управления и плана разработки | 08.10.2016 – 15.10.2016 |
| Разворачивание тестовой версии Moodle на локальном сервере | 22.10.2016 – 29.10.2016 |
| Согласование и утверждение технического задания |  |
| Разработка технического задания |  |
| Разработка пояснительной записки |  |
| Разработка описания используемых программных средств |  |
| Разработка мероприятий для внедрения системы |  |
| Приведение развернутой системы в соответствие требованиям |  |
| Верификация и тестирование системы |  |
| Внедрение системы |  |

* + 1. Выбор [языков программирования](javascript:void(23473))

В связи с тем, что разрабатываемая система базируется на системе управления курсами Moodle, в качестве основного будет использован язык PHP 5.6. Поскольку система представляет из себя веб-приложение, для создания и исправления шаблонов будет использован язык HTML 5, а для обеспечения интерактивности и выполнения некоторых функций системы – JavaScript.

* + 1. [Согласование](javascript:void(9745)) и [утверждение](javascript:void(9746)) [технического задания](javascript:void(11959))

После разработки технического задания его необходимо верифицировать. После прохождения процедуры верификации техническое задание необходимо утвердить.

1. [Технический проект](javascript:void(18925))

Разработка технического проекта

Данный пункт подразумевает разработку следующих документов: пояснительная записка, описание комплекса аппаратных и программных средств и спецификация оборудования и программных средств.

* + 1. Разработка [структуры](javascript:void(17183)) системы

В рамках разработки структуры системы необходимо выделить ее основные компоненты (подсистемы) и связи между ними, а также состав этих подсистем и выполняемые ими функции. Структура, т.е. подсистемы и связи между ними, могут быть выражены в виде диаграмм.

* 1. Утверждение технического проекта
     1. Разработка плана мероприятий по разработке и внедрению программ

Стадия ввода системы в действие включает в себя следующие этапы:

* Подготовка системы к вводу в действие.
* Подготовка существующей системы к объединению с разрабатываемой
* Подготовка стенда, на котором будет установлена система.
* Установка системы на стенд.
* Проведение предварительных испытаний.
* Проведение опытной эксплуатации.

Разработка, внедрение и эксплуатация системы осуществляется проектной документацией, учитывающей конкретные условия функционирования системы.

* + 1. Согласование и утверждение технического проекта

Осуществляется согласование и утверждение следующих документов: технического задания, пояснительной записки, планов верификации, конфигурационного управления и разработки. На рассмотрение, согласование и утверждение представляют документы технического проекта, прошедшие верификацию и отвечающие всем предъявленным к нему требованиям и стандартам.

1. Создание рабочего проекта
   1. Разработка программы

Разработка программы ведется в соответствии с планом-графиком, приведенным в начале данного документа. Возникающие задачи разделяются между всеми участниками команды в соответствии с их навыками и возможностями и ставятся с помощью сервиса Trello.

Задачи может создавать любой участник команды. При постановке задачи создается новая карточка в Trello в списке To Do. В названии карточки необходимо кратко описать задачу. В поле Description изложено более подробное описание задачи(если это требуется). Назначается крайний срок, до наступления которого задача должна быть выполнена. Одновременно заводится аналогичный Isuue в GiHub. Issue необходимо привязать к созданной в Trello карточке используя Addon;

Если у задачи из списка To Do отсутствует исполнитель, любой участник команды может указать себя в качестве исполнителя задачи. Если задача долго находится в списке To Do без указанного исполнителя - менеджер проекта самостоятельно назначает исполнителя;

Когда исполнитель начинает работать над задачей, он должен перенести ее из списка To Do в список In Progress. В своем fork’е основного репозитория на аккаунте GitHub необходимо сделать отдельную ветку(от ветки master) со следующим названием: “Issue[Номер Issue на основном репозитории на GitHub]” (например: Issue1);

Когда исполнитель задачи закончил работу над задачей, он помещает ее в список Testing. В сервисе GitHub необходимо сделать Pull Request своей ветки(напр.: Issue1) в ветку master основного репозитория;

Группа верификации и остальные разработчики должны выполнить верификацию кода и написать комментарии, если были обнаружены ошибки. Для исправления мелких ошибок исполнитель может обновлять свой Pull Request и отвечать на соответствующие комментарии(напр.: Исправлено). При наличии существенных ошибок или недоработок, карточка задачи переносится обратно в список In Progress. После исправления, разработчик обновляет свой Pull Request и переводит карточку задачи обратно в список Testing;

Если у группы верификации нет претензий или комментарий к Pull Request’у, то они самостоятельно переводят карточку задачи в список Done и выполняют Merge Pull Request’а.

* + 1. [Программирование](javascript:void(16832)) и [отладка программы](javascript:void(11446))

Написанный программный код должен соответствовать стандартам кода соответствующего языка. Требований к среде разработки не предъявляется.

* 1. Разработка программной документации
     1. Разработка программных документов

Необходимо разработать документацию двух типов: проектную — обзор программного обеспечения, включающий описание рабочей среды и принципов, которые должны быть использованы при создании ПО, и техническую — документацию на [код](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), [алгоритмы](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC), интерфейсы, [API](https://ru.wikipedia.org/wiki/API) итд. Разработка программных документов ведется в соответствии с календарным планом, приведенным выше, разработанные документы проходят верификацию. Документация должна храниться в общем репозитории для обеспечения ее версионного контроля.

* 1. Тестирование системы
     1. Разработка, согласование и утверждение [программы и методики](javascript:void(11958)) тестирования

Результатом данного пункта является план верификации и тестирования, разрабатываемый на основе взаимодействия группы разработчиков и группы верификации.

* + 1. Проведение тестирования

Проведение тестирования регламентируется пунктами «Проверка нового функционала» и «Функциональное тетирование нового фнкционала» плана верификации и тестирования.

* + 1. [Корректировка программы и программной документации](javascript:void(9688)) по [результатам](javascript:void(24714)) тестирования

Данный процесс связан с процессом, описанным в пункте «Регрессионное тестирование» плана верификации и тестирования. В случае нахождения проблем при проведении регрессионного тестирования команда разработчиков уведомляется об этом и приступает к исправлению кода.

1. [Внедрение](javascript:void(18927))
   1. Подготовка и передача системы
      1. Подготовка и передача системы и [программной документации](javascript:void(10666)) для [сопровождения](javascript:void(11449))

Данный этап подразумевает верификацию системы и всех пунктов документации, а также выполнение действий, необходимых для ее подготовки к внедрению в уже существующую систему.

* + 1. Передача программы заказчику

Заказчику должна быть передана работающая внедренная в использование система, а также полный комплект документов к ней.