Программная система учёта знаний учащихся в средней школе.

**Этап 1.**

1) Краткое словесное описание.

Темой данного курсового проекта является закрытая программная система учёта знаний учащихся в средней школе. Подобная программная система позволяет образовательным учреждениям расширить функциональные возможности бумажных журналов и дневников. Пользователями системы учета знаний учащихся станут родители и ученики (вынесены в роль «Ученик»), а также руководство школы: учителя, завучи и директор. За развёртывание и поддержку работоспособности системы в рамках данной предметной области отвечает системный администратор.

Программа предоставляет широкие возможности ученикам и их родителям. Пользуясь любым устройством, имеющим подключение к сети интернет, они могут просматривать оценки и домашние задания, получаемые в рамках учебного процесса. Ведение электронного дневника и журнала, помимо бумажных аналогов, позволит контролировать успеваемость, а также грамотно планировать расписание и выполнение домашней работы, что, в свою очередь, отразится на успешном выполнении учебного плана.

Пользуясь программной системой учета знаний персонал так же получит широкие возможности для планирования и контроля учебного процесса. В системе будут реализованы следующие функции: выставление оценок, составления домашнего задания и просмотр статистики. Возможность работы с журналом пользуясь любым устройством имеющего доступ к интернету, намного удобней, чем заполнение бумажного аналога в рамках одного занятия. Сформированная программной системе статистика позволит учителю наглядно представить насколько хорошо ученики класса усвоили какой-либо материал.

Администрация учебного заведения с помощью программной системы сможет добавлять учителей и учеников в базу данных, создавать списки новых и текущих классов и формировать для них удобное расписание. Система предусматривает две роли для администрации школы: заведующий учебной частью школы (завуч) и директор. При этом происходит разделения обязанностей. Функции добавления учеников, формирования классов и расписания будут доступны завучу, при этом директор получит возможность добавления персонала школы.

Системный администратор занимается развёртыванием и поддержкой программной системы в рамках какой-либо школы. Данный работник имеет все основные функции системы: добавление/редактирование/удаление списка пользователей, просмотр/изменения/удаление записей в журнале/дневнике, формирование/удаление классов и расписаний – что позволит администратору поддерживать актуальность справочников учеников и преподавателей, а также контролировать правильность заполнения журнала.

Регистрация новых пользователей производится персоналом школы.

Программная система учёта знаний учащихся будет представлять веб-приложение, созданное с использованием модели MVC, реляционной базы данных MySQL и реализована на языке C#. Данный выбор обусловлен тем, что логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо, что позволит максимально эффективно разрабатывать и поддерживать программную систему. Реляционная БД, в свою очередь, проста в использовании и имеет широкие возможности манипулирования имеющимися данными. Интеграции с какими-либо внешними сервисами не производиться.

2) Словарь предметной области с описанием объектов предметной области и   
их характеристика.

*Программная система* - программная система учёта знаний учащихся в средней школе.

*Гость* – любой неавторизированный участник программной системы. Не имеет доступа к каким-либо функциям.

*Пользователь* – какой-либо авторизированный участник программной системы. Каждый пользователь имеет единственную определенную роль.

*Справочник* – фрагмент базы данных, хранящий информацию об группе определенных объектов. Определенные группы пользователей могут выполнять редактирование/добавление этих справочников, а также информации в них.

*Класс* – совокупность учеников, выделенных в отдельную группу. Каждый класс имеет свое уникальное расписание.

*Электронный дневник* – предоставляет возможность просмотра оценок и домашнего задания ученика.

*Электронный журнал* – предоставляет функции просмотра справочника классов и учеников, а также редактировать информацию, представленную электронном дневнике.

*Ученик* – роль при которой пользователь, который имеет функцию просмотра своего электронного дневника, а также принадлежит к какому-либо классу. Регистрацию в системе производит завуч или директор.

*Учитель* – роль при которой пользователь, который имеет доступ как к электронному дневнику, так и электронному журналу. Регистрацию в системе производит директор.

*Завуч* – роль при которой пользователь, который имеет права учителя, а также функции просмотра и редактирования справочников учеников, классов и их расписания. Регистрацию в системе производит директор.

*Директор* – роль при которой пользователь, который имеет права завуча, а также функции просмотра и редактирования справочников учителей и завучей. Учетная запись уникальна и заведомо добавлена в базу данных программной системы.

*Администратор* – роль при которой пользователь имеет доступ ко всем функциям системы. Системный администратор школы, отвечающий за функционирование программной системы.

*Персонал* – пользователь, который имеет одну из следующих ролей: учителя, завучи, директор, администратор.

*Администрация* – пользователь, который имеет одну из следующих ролей: директор, завучи и администратор.

3) Перечень ролей, которые предусмотрены в системе:

* Гость
* Ученик
* Учитель
* Завуч
* Директор
* Администратор

4) Диаграмма прецедентов.

Диаграмма прецедентов представлена на рисунке 1.



Рисунок 1. Диаграмма прецедентов

5) Спецификация прецедентов.

* Название: «Просмотр дневника»

Предусловие: пользователь имеет доступ к веб-сайту программной системы.

Действующее лицо: пользователь

Основной поток: Действующее лицо открывает окно приложения, отображающую информацию, заполненную в дневнике. С помощью клавиш навигации осуществляет выборку необходимой информации. Прецедент при этом использует функции просмотра оценок и просмотра домашнего задания. Персонал может просматривать дневник любого ученика, выбрав его в выпадающем списке. При завершении работы действующее лицо закрывает окно приложения.

Альтернативный поток: Если действующее лицо желает произвести выборку по критериям, то система осуществляет запрос к БД и выдает необходимые данные.

* Название: «Управление журналом»

Предусловие: персонал имеет доступ к веб-сайту программной системы.

Действующее лица: персонал.

Основной поток: действующее лицо открывает форму электронного журнала и выполняет необходимые ему действия с помощью клавиш навигации, тем самым имея возможность активировать прецедент «редактирование журнала». В окне представлены справочники классов и учеников, их оценки и домашнее задание. При завершении работы действующее лицо закрывает окно приложения.

* Название: «Добавить ученика»

Действующее лица: администрация.

Основной поток: действующее лицо открывает форму заполнения информации об ученике. Выполняет корректный ввод данных в формы (выполняет ввод следующих полей: имя, фамилия, отчество, адрес и телефон строкового типа, выбирает в календаре дату рождения) выбирает класс из выпадающего списка и нажимает кнопку «Добавить». Система создает нового пользователя с ролью «ученик» и сохраняет данные в БД.

Альтернативный поток:

* Действующее лицо неправильно заполнило поля, при этом получил сообщение об ошибке. В таком случае, данный пользователь должен перепроверить и исправить введенные данные
* Действующее лицо не желает добавлять нового ученика по каким-либо причинам и нажимает кнопку «Отмена». Система не создает нового пользователя и не сохраняет введенные данные, происходит редирект на предыдущую страницу.
* Если введенные ключевые поля принадлежат уже зарегистрированному пользователю, система выдает сообщение об ошибке.

Этап №2.

Моделирование необходимо для понимания системы. При этом ни одна модель не является абсолютно достаточной. Напротив, чтобы понять большинство систем, кроме самых тривиальных, часто требуется множество взаимосвязанных моделей. В данном этапе будут представлены: *диаграмма классов, диаграмма состояния и диаграмма последовательностей*.

**Диаграмма классов.**

Сущности диаграммы:

Class (Класс)

* -Number (Номер класса)
* -Apprentice (Ученики класса)
* -Schedule (Расписание класса)
* -Note (Примечание)

Schedule (Расписание) – расписание, которое включает в себя занятия на каждый день недели.

* -Days : Array<Day\_of\_the\_week> (Дни недели)
* -Status : Schedule\_Status (Статус расписания)
* -Note : String (Примечание

Schedule\_Status (Статус расписания) – отображает этап разработки расписания.

* -In developing (В разработке)
* -Under consideration (Ожидает подтверждения)
* -Done (Готово)

Days\_of\_the\_week (Дни недели) – учебные дни, содержащие от 1 до 7 предметов.

* -Subjects : Array<Subject> (Предметы)

Subject (Предмет) – учебное занятие.

* -Name : String (Название предмета)

Mark (Оценка) – число оценивающее работу по определённому предмету.

* -Mark : int (Оценка)
* -Subject : Subject (Предмет)
* -DateTime: DateTime (Дата получения)

Homework (Домашнее задание) – задание ученика по определенному предмету.

* -Homework : String (Домашнее задание)
* -Subject : Subject (Предмет)
* -DateTime : DateTime (Дата получение)

Person (Персонал)

* -First\_Name : String (Имя)
* -Second\_Name : String (Фамилия)
* -Patronymic : String (Отчество)
* -Phone\_number : String (Телефон)
* -E\_mail : String (Электронная почта)

Apprentice (Ученик)

* -Class\_number : String (Номер класса)
* -Marks : Array<Mark> (Оценки ученика)
* -Homework : Array<Homework> (Домашнее задание)
* -Parents\_phone\_number : String (Номер родителей)

Staff (Персонал)

* -Beginning of work : Date (Дата началы работы)
* -End of work : Date (Дата конца работы)
* -Salary : Decimal (Зарплата)
* -Specialty : String (Специальность)
* -Status : Status (Статус)

Status (Статус) – статус работника.

* Working (Работает)
* Dismissed (Уволен)
* Holiday (В отпуске)

Teacher (Учитель)

* -Subjects : List<Subject> (Предметы)
* -Class : List<Class> (Классы)

Классы Head\_teacher (Завуч) и Director (Директор) не имеют уникальных полей и различаются только реализуемыми методами.

Программной система реализована по шаблону MVC ASP.NET, методы доступа и взаимодействия с базой данных будут реализованы в классах DAO. На данном уровне абстракции эти методы исключены из диаграммы.

Диаграмма классов представлена на рисунке 2.



Рисунок 2. Диаграмма классов.

Состояние объекта «Расписание».

1. Пользователь (завуч) составляет расписание, с помощью выпадающих списков выбирает необходимые предметы на каждый день недели. Объект «расписание» создан в системе (объект в памяти).

2. Производится либо отправка расписания, либо сохранение расписания со статусом «в разработке». Объект добавляется в БД с соответствующим статусом.

3. Если расписание находится в статусе «в разработке», возможно ее удаление, либо отправка на проверку.

4. После отправки, расписание переходит в статус «Ожидает подтверждения», при этом директор может отклонить или подтвердить расписание. В таком случае объект переходит в статус «в разработке» и «готово» соответственно. То есть объект обновляется в БД с соответствующим статусом.

Диаграмма состояния объекта расписание представлена на рисунке 3.



Рисунок 3. Диаграмма состояния.

Диаграмма последовательностей.

Завуч и директор являются актерами. ScheduleManager – класс ИС, реализующий подсистему работы с расписаниями. Schedule – расписание.

Методы CreateSchedule(), UpdateSchedule(), DeleteSchedule() имеют реализацию в классах DAO.

Диаграмма последовательностей представлена на рисунке 4.



Рисунок 4. Диаграмма последовательностей.