## Вариант 17: Необходимо разработать приложение для связи старост групп со службами университета.

В системе необходимо наличие ролей следующих типов:

- «Староста»

- «Деканат»

- «Преподаватель»

Необходимо реализовать следующий функционал для пользователя с ролью «Староста»:

- просмотр расписания группы.

- изменение расписания (при согласовании с «Преподавателем»);

- контроль посещаемости занятий и отправка отчётов пользователю с ролью «Преподаватель»

Необходимо реализовать следующий функционал для пользователя с ролью «Преподаватель»:

- изменение расписания (с автоматическим уведомлением пользователей типа «Староста», которых затрагивает это изменение);

- контроль посещаемости занятий

- назначение дат зачётов и переэкзаменовок.

- просмотр личного расписания.

- просмотр занятости аудиторий

- бронирование аудиторий при переносе занятия.

Необходимо реализовать следующий функционал для пользователя с ролью «Деканат»:

- просмотр и редактирование расписания с уведомлениями преподавателей и студентов;

- формирования списков студентов.

Необходимо также реализовать следующую логику: если меняется расписание, то после авторизации преподавателя или студента, расписание которого было изменено, ему должно отображаться оповещение об изменениях (например, в виде всплывающего окна).

1 Анализ предметной области

1.1 Расширение описания варианта

Отдельно выделим следующие объекты:

* Группа (студентов)

База данных должна содержать информацию о пользователях(ФИО, группа, подгруппа, статус, роль, контактный телефон) группах (название, состав, расписание), расписании (группа, время, аудитория, преподаватель, тип занятий), аудиториях( номер, статус) .

В системе есть такие роли как:

- «Староста»

- «Деканат»

- «Преподаватель»

Необходимо реализовать отдельный функционал для пользователя с ролью «староста»:

- просмотр расписания группы.

- изменение расписания (при согласовании с «Преподавателем»);

- контроль посещаемости занятий и отправка отчётов пользователю с ролью «Преподаватель»

Необходимо реализовать следующий функционал для пользователя с ролью «Преподаватель»:

- изменение расписания (с автоматическим уведомлением пользователей типа «Староста», которых затрагивает это изменение);

- контроль посещаемости занятий

- назначение дат зачётов и переэкзаменовок.

- просмотр личного расписания.

- просмотр занятости аудиторий

- бронирование аудиторий при переносе занятия.

Необходимо реализовать следующий функционал для пользователя с ролью «Деканат»:

- просмотр и редактирование расписания с уведомлениями преподавателей и студентов;

- формирования списков студентов.

Необходимо также реализовать следующую логику: если меняется расписание, то после авторизации преподавателя или студента, расписание которого было изменено, ему должно отображаться оповещение об изменениях (например, в виде всплывающего окна).

1.2 Формирование бизнес-правил

* Уведомления
  + Могут приходить в виде:
    - Почты
    - Всплывающего уведомления
* Разные аудитории способны вмещать в себя различное количество студентов и групп.
* У пользователя может быть лишь одна пара в одновременно.
* У группы может быть лишь 1 пользователь с ролью «староста»
* У одного преподавателя и студента может быть несколько пар в день.