Федеральное государственное бюджетное образовательное учреждение

высшего образования (ФГБОУ ВО)

«Дальневосточный государственный университет путей сообщения»

Кафедра: «Информационные технологии и

системы»

Лабораторная работа № 4

Тема: «Разработка диаграммы компонентов»

Выполнил: Козеко М.В.

гр.СО251КОБ

Проверил: Анисимов В.В.

Сазанова Е.В.

Хабаровск, 2020

**Задание на лабораторную работу**

Разработать диаграммы компонентов по индивидуальному заданию (см. Выбор задания на выполнение лабораторных работ):

- диаграмму, специфицирующую структуру исходного кода;

- диаграмму, специфицирующую состав компонентов на рабочем месте пользователя.

**Ход выполнения работы**

Диаграмма компонентов описывает особенности физического представления системы.

Она позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный и исполняемый код.

Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

Диаграмма компонентов разрабатывается для следующих целей:

* визуализации общей структуры исходного кода программной системы;
* спецификации исполняемого варианта программной системы;
* обеспечения многократного использования отдельных фрагментов программного кода;
* представления концептуальной и физической схем баз данных.

В языке UML для компонентов определены следующие стереотипы:

* «file» – любой файл, кроме таблицы;
* «executable» – программа (исполняемый файл);
* «library» – статическая или динамическая библиотека;
* «source» – файл с исходным текстом программы;
* «document» – остальные файлы (например, файл справки);
* «table» – таблица базы данных.

На рисунке 1 показана диаграмма компонентов, в которой отражены основные элементы разрабатываемой информационной системы, а также основной исполняемый файл, файлы веб-форм приложения, служебные файлы с дополнительной информацией, а также база данных с соответствующими интерфейсами.

Как видно из рисунка 1, при реализации программного продукта будет использоваться концепция MVC (Model-View-Controller). Согласно данной концепции, программный продукт будет разделен на три составные части, которые будут взаимодействовать друг с другом.

Каждая из логических частей концепции представлена в соответствующей папке: Models (здесь хранятся классы модели), Controllers (здесь хранятся контроллеры) и Views (здесь хранятся представления в форме cshtml кода).

Папка Models (Модели) содержит в себе 8 классов: community (сообщество), member (участники), address (адрес), friends (друзья), directory\_worktime (справочник смен), specialist (технический специалист), user (пользователь), request(обращение), которые были описаны на диаграммах классов приложения. Каждый из этих классов может связываться с базой данных SocialWeb при помощи ADO.NET

Папка View (Представления) разбита на подпапки в соответствии с тем, какому контроллеру принадлежат представления в данной вложенной папке. Содержание папки View:

* Master - папка, содержащая в себе представления, позволяющие задать содержание мастер-страницы и другие дополнительные представления:
  + Main.cshtml – основная мастер-страница;
  + Login.cshtml - компонент, который отображает имя авторизированного пользователя в верхнем правом углу страницы;
* Specialist\_ac– папка, содержащая в себе представления, которые позволяют техническому специалисту работать с обращениями:
  + View-request.cshtml – страница просмотра обращений;
  + Edit\_request.cshtml – страница редактирования обращения;
* Admin\_ac – папка, содержащая в себе представления, которые позволяют администратору системы просматривать статистику сайта и информацию о системе. Так же имеется представление, для добавления нового работника.
  + Statistics.cshtml – страница просмотра статистики сайта;
  + Information.cshtml – страница просмотра информации о системе;
  + New\_worker.cshtml – страница добавления работника;
* Home – папка, содержащая в себе основные представления сайта:
  + Index.cshtml – главная страница сайта;
  + Autorization.cshtml – страница авторизации;
  + Registration.cshtml – страница регистрации;
  + Inform.cshtml – страница просмотра информации о сайте и местонахождении главного офиса;
* User\_ac – папка, содержащая в себе основные представления пользователя при работе с социальной сетью:
  + View\_profile.cshtml – главная страница профиля пользователя;
  + User\_page.cshtml – страница редактирования главной страницы пользователя;
  + User\_profile.cshtml – страница редактирования профиля пользователя;
  + View\_community.cshtml – страница просмотра всех сообществ;
  + Message.cshtml – страница просмотра сообщений;
  + Dialog.cshtml – страница диалога между двумя пользователями;
  + New\_request – страница создания нового обращения;
  + View\_request – страница просмотра обращений;
  + New\_community.cshtml – страница создания сообщества;
  + Help.cshtml – страница просмотра раздела “Помощь”;
  + Friends.cshtml – страница просмотра друзей;
  + Friends\_search.cshtml – страница поиска друзей;

Папка Controllers (Контроллеры) содержит классы контроллеров, которые обрабатывают запросы, управляют взаимодействием между моделью и представлением, выполняют ключевые вычислительные операции. Всего необходимо 5 контроллеров – по одному на каждую группу преставлений:

* user\_ac\_con.cs – контроллер для управления группой представлений «user\_ac»;
* admin\_ac\_con.cs – контроллер для управления группой представлений «admin\_ac\_con»;
* master\_ac\_con.cs – контроллер для управления группой представлений «master\_ac»;
* home\_con.cs – контроллер для управления группой представлений «home»;
* specialist\_ac\_con.cs – контроллер для управления группой представлений «specialist\_ac»;

На рисунке 2 представлена диаграмма компонентов серверного приложения, на которой отображены веб сервер, исполняемый файл, база данных, СУБД и ADO.NET при помощи которого исполняемый файл (приложение) подключается к СУБД.

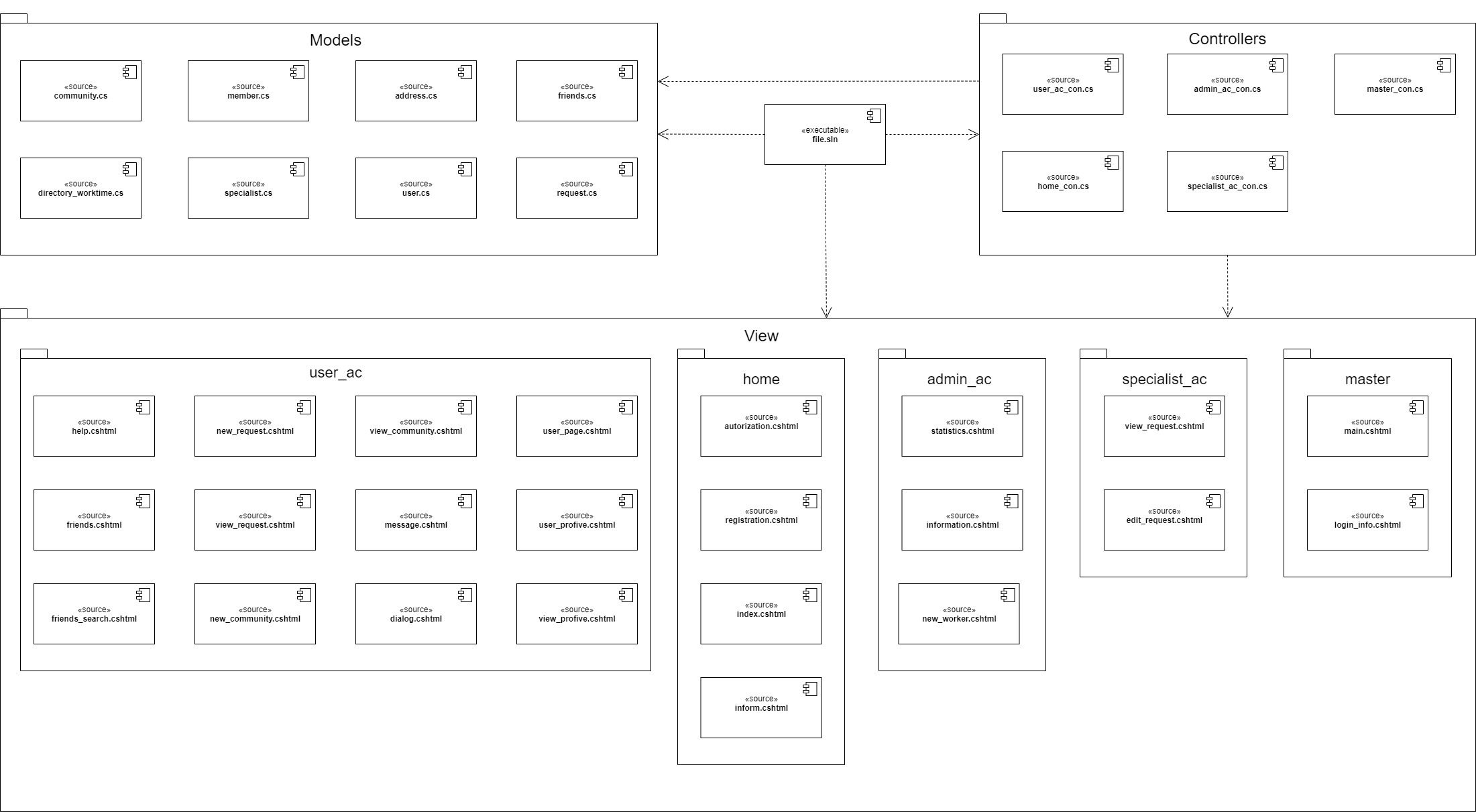


Рисунок 1 Диаграмма компонентов

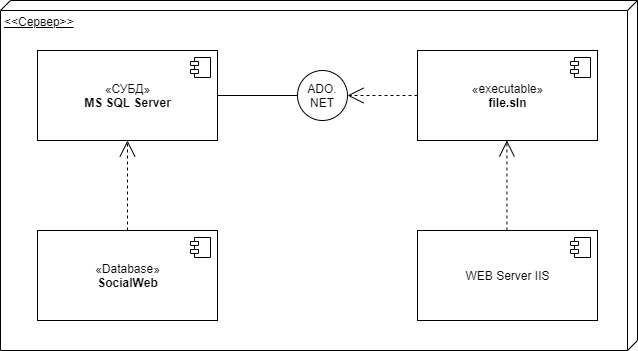


Рисунок 2 Диаграмма компонентов серверного приложения