Физическое представление программной системы не может быть полным, если отсутствует информация о том, на каких вычислительных средствах она реализована. Для представления общей конфигурации и топологии распределенной программной системы предназначены диаграммы развертывания. Диаграмма развертывания предназначена для визуализации элементов и компонентов программы. При этом представляются только компоненты-экземпляры программы, являющиеся исполняемыми файлами или динамическими библиотеками. Диаграмма развертывания содержит графические изображения процессоров, устройств, процессов и связей между ними.

При разработке диаграммы развертывания преследуются следующие цели:

1. определить распределение компонентов системы по ее физическим узлам;
2. показать физические связи между всеми узлами реализации системы на этапе ее исполнения;

В качестве элементов диаграммы используются: аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), программные компоненты («Artifact») работают на каждом узле (например, веб-приложение, база данных), различные части этого комплекса соединяются, друг с другом изображаются отрезками линий без стрелок. Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Существует два типа узлов: узел устройства («Device»), узел среды выполнения (execution environment).

Опишем несколько представлений программной системы разрабатываемой в рамках дипломного проекта.

На первой диаграмме развертывания  покажем связь и работу, которая реализована на данный момент:

На диаграмме показано, что программная система состоит из сервера базы данных (DataBaseServer) и рабочей станции, которая имеет подключение к данному серверу. Справа представлена рабочая станция, на которой инсталлировано и выполняется приложение podsystem.exe, которое является приложением, разработанным в дипломном проекте. Слева представлен узел устройства – сервер базы дынных, на котором развернуто такая среда выполнения как СУБД PostgreSQL, на которой в свою очередь может быть развернута любая база данных со своей группой таблиц (Group tables of DBMS) и двумя таблицами, таких как: таблица настроек (Table Settings), таблица записей аудита (Table Logs); которые создаются приложением podsystem.exe и инсталлируются в базу данных, к которой приложение имеет подключение, а также с триггером (Trigger Log), генерируемым в приложении podsystem.exe и также инсталлируемым в базу данных, как и две выше описанных таблицы.

Общий смысл диаграммы в том, что на рабочей станции имеется приложение podsystem, которое подключается к базе данных, находящейся под управлением системы PostgreSQL и располагающейся на каком-то сервере базе данных, и добавляет в эту базу данных необходимые таблицы, а так же триггер со своими параметрами для аудита.

Данная диаграмма представлена на Рисунке 1.

На второй диаграмме развертывания  покажем связь и работу при более широком возможном использовании данного проекта:

На диаграмме представлена расширенная версия первой диаграммы. Дополнение: есть узел устройства веб-сервер (WebServer), на котором развернута некая среда выполнения приложений (Application) и есть сервер приложений (ApplicationServer), между этими двумя серверами установлено соединение. Сам сервер приложений может использовать базу данных на сервере базы данных, по также установленному между ними соединению.

Общий смысл диаграммы в том, что также имеется рабочая станция, подключение к БД, сервер БД, как и в предыдущей диаграмме. Добавляется только возможность использования сервера баз данных неким другим сервером приложений. То есть можно предположить существование некоего веб-сервера, на котором установлены свои приложения, обслуживающие его работу, и работающие с сервером приложений, приложения которого могут использовать базу данных на сервере БД.

Данная диаграмма представлена на Рисунке 2.

На третьей диаграмме развертывания  покажем связь и работу при более широком возможном использовании проекта из второй диаграммы:

На диаграмме представлена расширенная версия второй диаграммы. Дополнение: есть ещё один узел устройства - сервер баз данных (DataBaseServer2), на котором, как и на первом сервере БД, развернута среда выполнения - СУБД PostgreSQL, на которой в свою очередь может быть развернута любая база данных со своей группой таблиц (Group tables of DBMS2) и двумя таблицами: таблица настроек (Table Settings2), таблица записей аудита (Table Logs2); которые создаются приложением podsystem и инсталлируются в базу данных, к которой приложение имеет подключение, а также с триггером (Trigger Log2), генерируемым в приложении podsystem и также инсталлируемым в базу данных, как и две выше описанных таблицы.

Общий смысл диаграммы в том, что также имеется рабочая станция, подключение к БД, сервер БД, сервер приложений, веб-сервер, как и в предыдущей диаграмме. Добавляется только возможность подключения приложения podsystem ко второй базе данных, также под управлением системы PostgreSQL и располагающейся на другом сервере базе данных. Приложение podsystem также добавляет в эту базу данных необходимые таблицы и триггер со своими параметрами для аудита.

В данном проекте возможность работы с несколькими базами данных не реализована, но в дальнейшем может быть добавлена. Данное добавление значительно расширяет функциональность проекта.

Данная диаграмма представлена на Рисунке 3.

На четвертой диаграмме развертывания  покажем связь и работу при самом широком возможном использовании проекта:

На диаграмме представлена расширенная версия третьей диаграммы. Дополнение: есть соединение между узлом - сервер приложений (ApplicationServer) и узлом - сервер баз данных (DataBaseServer2). Сервер приложений может использовать, по установленным между ними соединению, обе базы данных на сервере базы данных и сервере базы данных 2.

Общий смысл диаграммы в том, что также имеется рабочая станция, подключение к БД, сервер БД, сервер приложений, веб-сервер, возможность подключения приложения podsystem ко второй базе данных, также под управлением системы PostgreSQL и располагающейся на другом сервере базе данных. Добавляется возможность использования приложениями с сервера приложений второй базы данных, таким ж образом как это происходит в случае работы приложения podsystem.

Данная диаграмма представлена на Рисунке 4.

Далее рассмотрим диаграммы деятельности для разрабатываемого проекта.

По диаграмме видно, что после запуска программы происходит операция соединение с СУБД, данное соединение происходит благодаря заранее созданному драйверу PostgreSQL. Далее следует символ принятия решения – если соединение не установлено, то дальнейшая работа приложения невозможна, и оно приходит к завершению, если же соединение установлено, то переходим к следующему этапу – запросу к БД. Операция запроса БД определяет, есть ли уже в БД необходимые для работы приложения таблицы. Далее идет распараллеливание, а именно одновременно выполняется две операции: 1) в зависимости от результата предыдущей операции принимается решение, если таблицы существуют, то происходит переход к следующей операции, если нет, то необходимые таблицы, а это таблицы настроек и таблица записей аудита, создаются в БД. 2) осуществляется операция ввода пароля и пользователя и дальнейшее определения прав этого пользователя. Далее происходит слияние потоков управления и создается форма приложения в которую загружается таблица записей аудита, если она была создана на предыдущем этапе, то пустая, если с ней была произведена какая-то работа, то с набором записей. В зависимости от того какие права были определены на предыдущем этапе для данного пользователя далее поток опять разветвляется. Если у данного пользователя имеются права только на чтение, то для него следует операция чтения без редактирования таблицы записей аудита и без изменения и чтения настроек, после чего возможен только выход из приложения. Если ж для пользователя определены все права, то переходим к следующей операции-генерации триггер, которая будет рассмотрена более подробно отдельно. Параллельно операции генерации, либо после нее возможна операция чтение и редактирование таблицы настроек, а также установка фильтров для просмотра таблицы записей аудита, которые находятся на форме приложения. После этих операций возможен либо выход, либо их неоднократный повтор. Операция генерации триггера: генерация триггера начинается с выбора таблицы, далее выполняется операция проверки на существование в БД триггера на выбранную таблицу, поскольку создание нескольких триггерных функций на одну таблицу невозможна; После этого происходит ветвлении на две операции: 1) если есть такая триггерной функция, то появляется сообщение об этом 2) если нет, то переходим к следующему этапу. Следующий этап заключается в операции выбора полей для выбранной таблицы. Потом можно добавить комментарии для выбранных полей или же оставить их пустыми. Также возможно дальнейшая операция установки области действия триггера и установки операций на которые будет срабатывать данный триггер. После всех этих операция происходит распараллеливание. Параллельно выполняется операция записи триггера в БД и операция записи в таблицу настроек выбранных параметров (выбранные поля, выбранная таблица, имя пользователя, который совершил все операции при генерации). На этом генерация заканчивается.