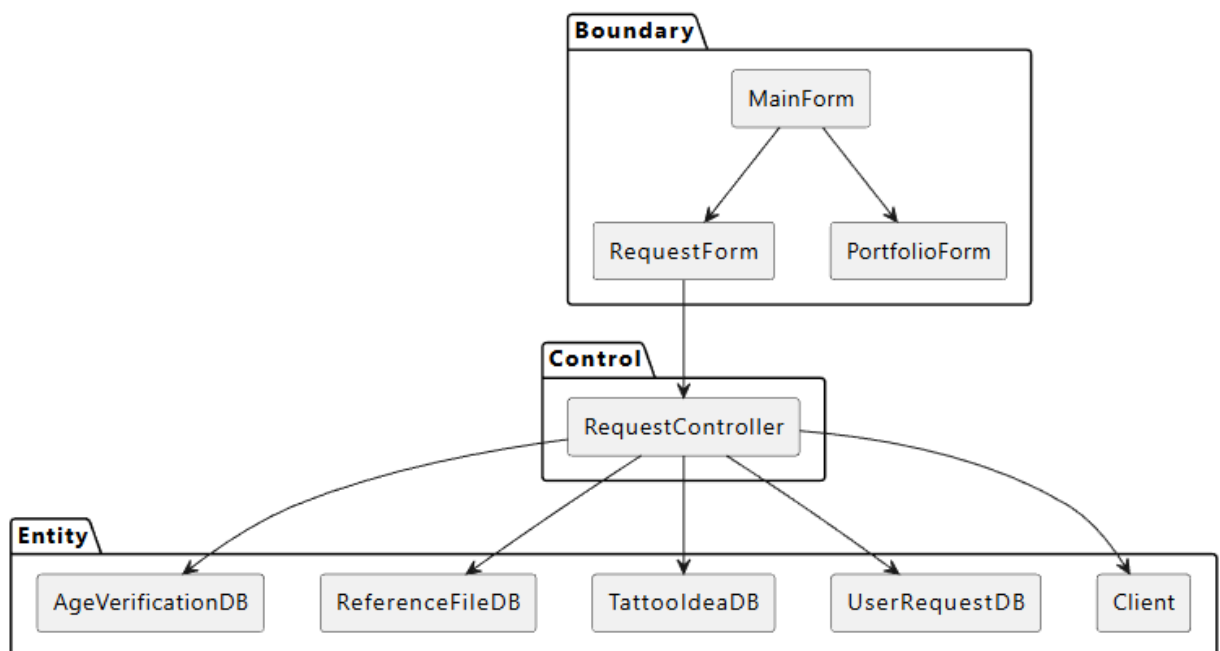


## ГЛАВА 6. ОРГАНИЗАЦИЯ ТЕСТИРОВАНИЯ И ПРИЁМКИ ИНФОРМАЦИОННОГО РЕСУРСА

Одно из свойств Enterprise Architect — возможность генерации программного кода, реализующего созданную модель. В данной главе будет показана разработка диаграммы компонентов, из которой также будет сгенерирован программный код классов на C++.

После формирования диаграммы компонентов она была разделена на три пакета: entity, boundary и control. Разбиение диаграммы по пакетам представлено на рисунке 6.1.



**Рисунок 6.1 – Схема данных**

Источник: собственная разработка

Впоследствии диаграмма компонентов была декомпозирована. Ниже представлены пакеты boundary (рисунок 6.2), entity (рисунок 6.3) и control (рисунок 6.4).

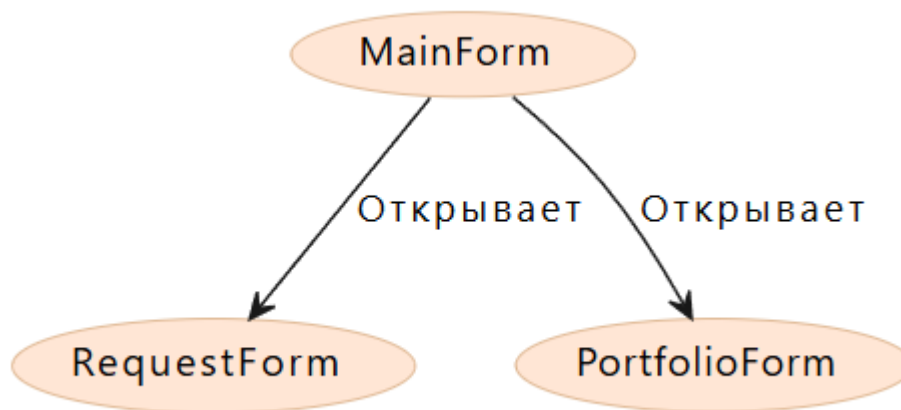


Рисунок 6.2 – Пакет Boundary

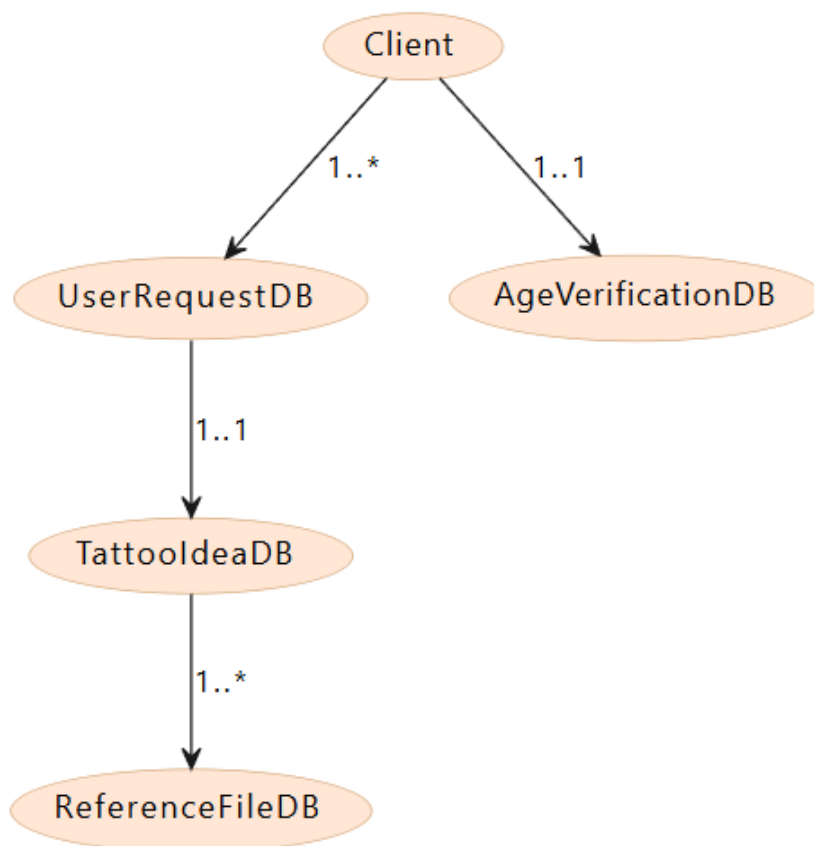


Рисунок 6.3 – Пакет Entity

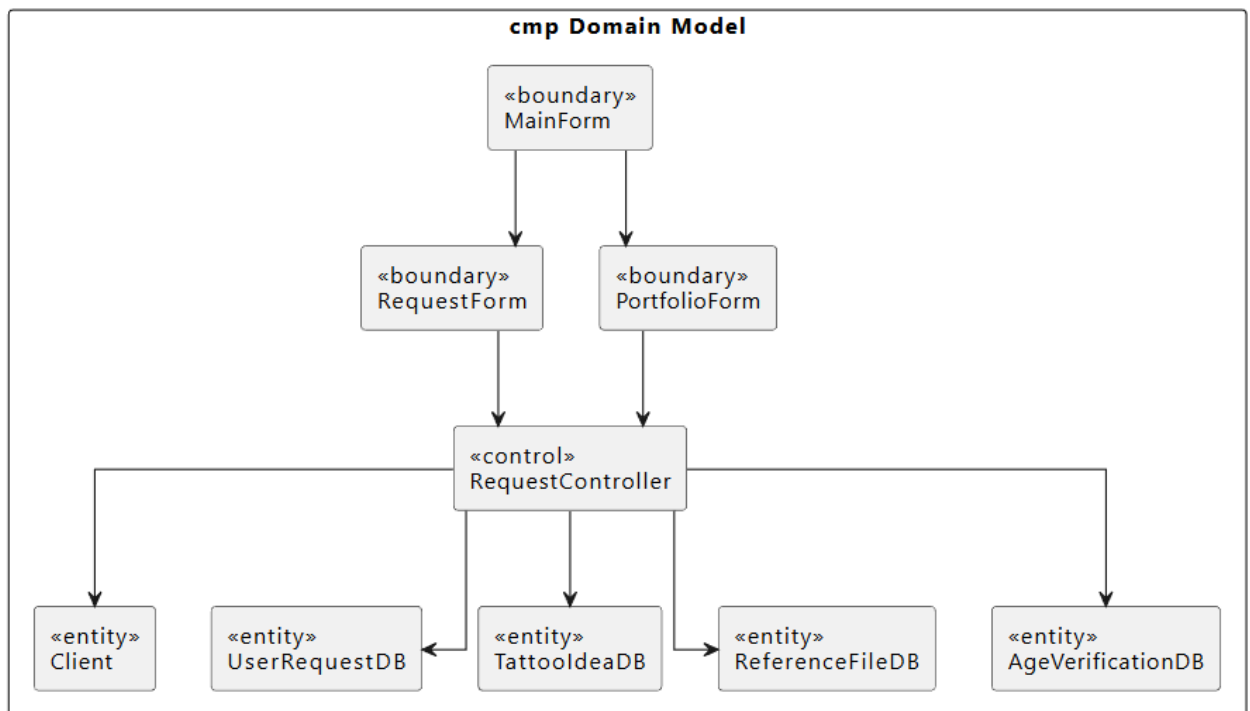
Источник: собственная разработка



**Рисунок 6.4 – Пакет Control**

Источник: собственная разработка

На рисунке 6.5 представлена итоговая диаграмма компонентов.



**Рисунок 6.5 – Диаграмма компонентов**

Источник: собственная разработка

После формирования диаграмм были сгенерированы файлы с кодом для каждого класса посредством внутреннего инструментария среды Enterprise Architect. Ниже приведён пример сгенерированного шаблона класса RequestForm.

```
#include "Client.h"
```

```
#include "TattooIdeaDB.h"
```

```
#include "ReferenceFileDB.h"
```

```
#include "RequestController.h"
```

```
class RequestForm
```

```
{
```

```
public:
```

```
    RequestForm();
```

```
    virtual ~RequestForm();
```

```
    void open();
```

```
    void get_info();
```

```
    void save();
```

```
    void set_info();
```

```
private:
```

```
    RequestController *m_Controller;
```

```
    Client m_Client;
```

```
    TattooIdeaDB m_Idea;
```

```
};
```