Пользователь нажимает "New project" и выбирает наш тип модуля.

На странице свойств модуля выпадающий список "Project template". В списке один элемент: "Architecture - с4 short".

Выбираем элемент, появляются поля имён файлов:

* Architecture file name: architecture.adoc
* Context diagram file name: c4context.drawio

Имена файлов нельзя изменять.

На втором этапе можно будет добавить возможность изменять файлы, другие шаблоны (например, "Archtecture - с4 full" с двумя диаграммами context и containers) и т.д.

В соответствии с выбранным шаблоном в проект добавляется:

* /.work/architecture-template.adoc
* /diagrams/c4context.drawio
* /architecture.adoc

Файлы c4context.drawio и architecture.adoc пустые. Шаблон architecture-template.adoc представляет из себя текст в формате AsciiDoc с элементами шаблонизатора Apache FreeMarker.

Текст шаблона:

= Архитектура решения

:toc:

:toc-title: Содержание

== Глоссарий

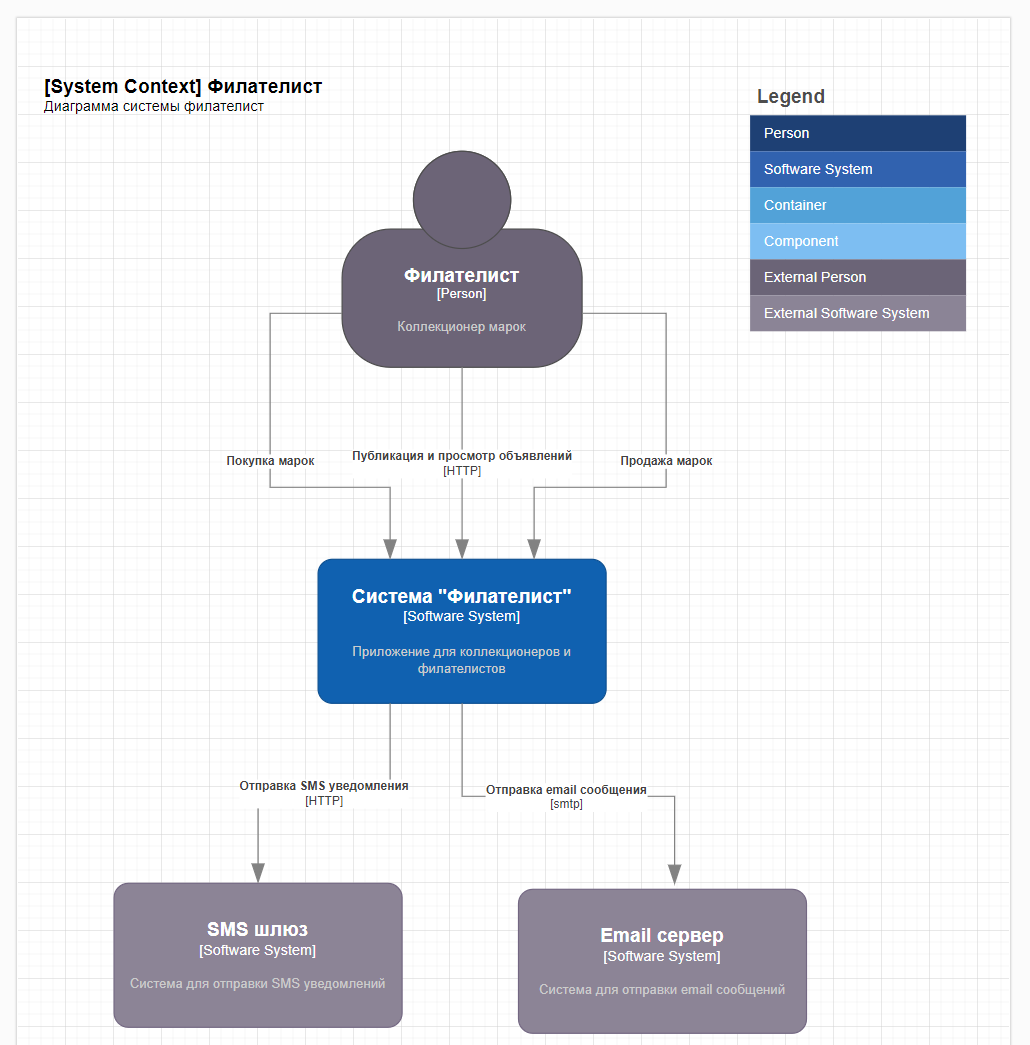
${Terms}

== Контекст решения

${System-Context-Image}

${System-Context-Description}

Рисуем диаграмму, сохраняем.



Для генерации описания будет использоваться следующая часть xml-ины диаграммы:

<mxfile>

<diagram id="7EPNmoJp4E522ICIcQfR">

<mxGraphModel>

<root>

<object c4Name="Филателист" **c4Type="Person"** c4Description="Коллекционер марок"id="lcF5Oina7H7nnB5ik\_3u-17"></object>

<object c4Name="Система &quot;Филателист&quot;" **c4Type="Software System"** c4Description="Приложение для коллекционеров и филателистов" id="SipI1-Q3VBLqGM\_P5yqC-2">

<mxCell style="**fillColor=#1061B0**;"></mxCell>

</object>

<object **c4Type="Relationship"** c4Technology="HTTP" c4Description="Публикация и просмотр объявлений" id="SipI1-Q3VBLqGM\_P5yqC-3">

<mxCell **source="lcF5Oina7H7nnB5ik\_3u-17" target="SipI1-Q3VBLqGM\_P5yqC-2"**></mxCell>

</object>

<object c4Name="Email сервер" **c4Type="Software System"** c4Description="Система для отправки email сообщений" id="dNJ846vaSPyjx3agIrUi-1">

<mxCell style="**fillColor=#8C8496**"></mxCell>

</object>

<object c4Name="[System Context] Филателист" **c4Type="ContainerScopeBoundary"** c4Description="Диаграмма системы филателист" id="dNJ846vaSPyjx3agIrUi-2">

</object>

<object **c4Type="Relationship"** c4Technology="smtp" c4Description="Отправка email сообщения" id="dNJ846vaSPyjx3agIrUi-3">

<mxCell **source="SipI1-Q3VBLqGM\_P5yqC-2" target="dNJ846vaSPyjx3agIrUi-1"**></mxCell>

</object>

<object c4Name="SMS шлюз" **c4Type="Software System"** c4Description="Система для отправки SMS уведомлений" id="dNJ846vaSPyjx3agIrUi-4">

<mxCell style="**fillColor=#8C8496**;"></mxCell>

</object>

<object c4Type="Relationship" c4Technology="HTTP" c4Description="Отправка SMS уведомления" id="dNJ846vaSPyjx3agIrUi-5">

<mxCell source="SipI1-Q3VBLqGM\_P5yqC-2" target="dNJ846vaSPyjx3agIrUi-4"></mxCell>

</object>

<object c4Type="Relationship" c4Description="Покупка марок" id="dNJ846vaSPyjx3agIrUi-7">

<mxCell source="lcF5Oina7H7nnB5ik\_3u-17" target="SipI1-Q3VBLqGM\_P5yqC-2"></mxCell>

</object>

<object c4Type="Relationship" c4Description="Продажа марок" id="dNJ846vaSPyjx3agIrUi-9">

<mxCell source="lcF5Oina7H7nnB5ik\_3u-17" target="SipI1-Q3VBLqGM\_P5yqC-2"></mxCell>

</object>

</root>

</mxGraphModel>

</diagram>

</mxfile>

Пользователь нажимает Build.

Алгоритм обработки:

* Парсим файл /diagrams/proj-c4context.drawio как xml
* Выбираем все тэги у которых атрибут c4Type = "Person" или "Software System"
* Собираем найденные тэги в dto с атрибутами: type, name, description, external
* Для элементов с типом "Software System" заполняем св-во external. Если цвет элемента синий - external=false, иначе true.
* Устанавливаем связи между элементами, основываясь на тэгах с атрибутом c4Type="Relationship".
* Формируем текстовый запрос к нейросети для получения описания диаграммы (см. ниже). Отправляем запрос нейросети, получаем ответ.
* Формируем текстовый запрос к нейросети для получения списка терминов и сокращений (см. ниже). Отправляем запрос нейросети, получаем ответ.
* Формируем картинку для диаграммы. Сохраняем картинку в «/.work/system\_context\_{имя файла drawio}.jpeg».
* Формируем список параметров для подстановки в шаблон описания архитектуры. Список состоит из трёх параметров: Terms, System-Context-Image, System-Context-Description. Из шаблона «/.work/architecture-template.adoc» и параметров формируем финальный текст. Сохраняем его в «/architecture.adoc».

**Запрос к нейросети для получения описания диаграммы.**

Исходим из того, что на диаграмме одна система, ноль и более внешних систем, ноль и более пользователей.

Шаблон текстовки:

Подготовь описание для приложения {x} без преимуществ и недостатков. Приложение {x} предоставляет пользователю следующие функции: {x-action1, x-action2, …}. Приложение {x} использует внешние системы: {z1, z2, ...}. Внешняя система {z1} используется для {y1-action}. Внешняя система {z2} используется для {y2-action}. И т.д.

Параметры:

* x = name системы (элемента у которого c4Type = "Software System" и external = false)
* y = name пользователя (элемента у которого c4Type = "Person")
* z = name внешней системы (элемента у которого c4Type = "Software System" и external = true)
* x-action = description стрелочки из юзера в систему (c4Type="Relationship" + source=пользователь + target=система)
* y-action = description стрелочки из системы во внешнюю систему (c4Type="Relationship" + source=система + target=внешняя система)

Значения параметров берём из xml файла диаграммы.

Для примера выше получаем:

Подготовь описание для приложения Система "Филателист" без преимуществ и недостатков. Приложение Система "Филателист" предоставляет пользователю следующие функции: публикация и просмотр объявлений, покупка марок, продажа марок. Приложение Система "Филателист" использует следующие внешние системы: SMS шлюз, Email сервер. Внешняя система SMS шлюз используется для «Отправка SMS уведомления». Внешняя система Email сервер используется для «Отправка email сообщения».

Ответ от нейросети:

Описание приложения "Система "Филателист"":

Приложение предназначено для пользователей, увлеченных коллекционированием и продажей почтовых марок.

Основные функции:

Публикация и просмотр объявлений о продаже и покупке почтовых марок;

Осуществление покупок и продаж почтовых марок между пользователями.

Для реализации функциональности приложение интегрируется со следующими внешними сервисами:

SMS шлюз для отправки SMS уведомлений пользователям;

Email сервер для рассылки email сообщений.

Приложение позволяет объединить всех любителей филателии в едином сервисе, облегчив обмен информацией о наличии и продаже почтовых марок.

**Запрос к нейросети для получения списка терминов и сокращений**

Найди IT термины в следующем тексте. Напечатай найденный список терминов с кратким описанием на русском языке без итого. Текст:

"${neural\_response}"

Где ${neural\_response} – описание диаграммы, полученное от нейросети как есть без изменений

Для примера выше ответ такой:

Приложение - компьютерная программа, предназначенная для выполнения определенных функций на устройствах пользователей.

Функции - действия или операции, выполняемые программным обеспечением.

Интегрировать - объединять компоненты или системы.

SMS шлюз - устройство для передачи SMS сообщений между системами.

Email - электронная почта.

Email сервер - компьютер для приёма, хранения и отправки электронной почты.

Внешний сервис - система, предоставляющая внешние функции другому программному обеспечению.