**Problem Analysis**

**Content**

[**Statement** 2](#_Toc21281030)

[**Functional Requirements** 3](#_Toc21281031)

[**Object diagram** 5](#_Toc21281032)

[**Traceability** 6](#_Toc21281033)

[**Flowcharts** 8](#_Toc21281034)

[FR1 8](#_Toc21281035)

[FR3 9](#_Toc21281036)

[FR4 10](#_Toc21281037)

[FR5 11](#_Toc21281038)

[FR7 12](#_Toc21281039)

# **Statement**

Due to the growing awareness of the effects of human beings on the environment, various entities have begun to worry about helping to reduce the negative impact we have on the environment. Among those entities we find RECYCLES, the recycling staff wants to create an application that helps them train their employees and people in general on the proper management of waste, always trying to apply the 3R, Reuse, Recycle and Reduce.

The first thing you want to save is the information of the waste, of each waste you must indicate an identifier, a name, a source (industrial, residential, municipal, construction, hospital), a color, the time it takes to decompose (in days) and a product that produces it. There are different types of waste, and they are classified into different categories, the main categories are biodegradable, inert and recyclable. In addition to the characteristics of the aforementioned residues, the biodegradable ones want to know if it is suitable for composting or not. Recyclables want to know their type, the types can be: paper, cardboard, glass, plastics, metals. In addition to the type, it is desired to keep a description of what is the most appropriate way to make the disposition of these elements for homes and for industry. Inert residues are those that cannot be recycled and that take a long time to decompose, for example the icopor, of this type of waste you want to keep tips to reduce their use.

For each product associated with a waste, you want to keep an identifier, a name, and a description. It is expected that the system has the ability to search for information associated with a waste either by name or by the identifier of the product that produces it, for which the user must be shown a list of registered products. Keep in mind that a product can produce more than one residue.

|  |  |
| --- | --- |
| **Procedencia** | **Factor** |
| Industrial | 10% |
| Domiciliary | 5% |
| Building | 8% |
| Municipal | 12% |
| Hospitallers | 15% |

All waste has the ability to calculate its harmful effect on the planet taking into account its origin, to make this calculation multiply the time it takes to decompose by a factor determined by its origin according to the following table:

All residues perform this calculation, however, if it is a recyclable waste, 2% is subtracted and if it is a biodegradable waste suitable for composting, 1% is subtracted.

Biodegradable and recyclable waste has an additional behavior regarding whether it is usable or not. To calculate if a biodegradable waste is usable, it must be met that the decomposition time is less than one year and is suitable for composting. For recyclable waste, it must be met that there is a description of the proper way to dispose of the waste.

# **Functional Requirements**

|  |  |
| --- | --- |
| **NAME** | FR1 |
| **DESCRIPTION** | Allow to add a residue taking into account the type of waste, since it makes no sense to add a residue of any kind. It must be associated with a product that may or may not exist previously, if it does not exist it must allow the product to be added. |
| **INPUTS** | id, name, origin, color, decomposition days, product |
| **OUTPUTS** | Operation’s result message |

|  |  |
| --- | --- |
| **NAME** | FR2 |
| **DESCRIPTION** | Generate a report of the waste that is registered, where a title with the type of waste is displayed and the information of the waste of that type is listed. |
| **INPUTS** | Wastes array |
| **OUTPUTS** | Report string |

|  |  |
| --- | --- |
| **NAME** | FR3 |
| **DESCRIPTION** | Allow to add a product and the waste it can generate. |
| **INPUTS** | id, name, description |
| **OUTPUTS** | Operation’s result message |

|  |  |
| --- | --- |
| **NAME** | FR4 |
| **DESCRIPTION** | Find the information of a waste by waste name |
| **INPUTS** | waste name |
| **OUTPUTS** | Waste’s information if found; otherwise, report. |

|  |  |
| --- | --- |
| **NAME** | FR5 |
| **DESCRIPTION** | Find the information of wastes by product id |
| **INPUTS** | Product id |
| **OUTPUTS** | Information of wastes (notice the plural reference) if found; otherwise, report. |

|  |  |
| --- | --- |
| **NAME** | FR6 |
| **DESCRIPTION** | Show a list of registered products |
| **INPUTS** | None |
| **OUTPUTS** | Information of registered products. |

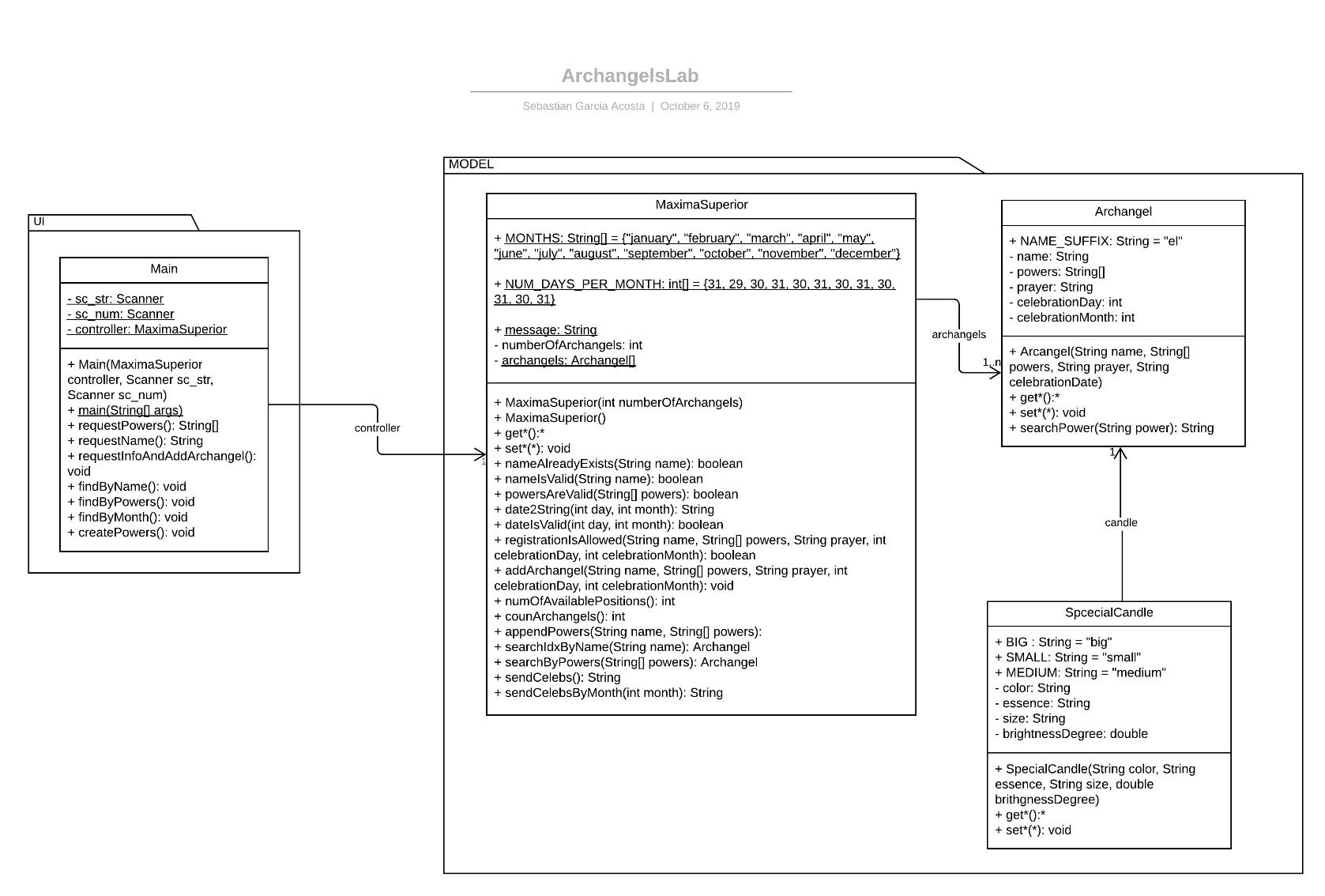
|  |  |
| --- | --- |
| **NAME** | FR7 |
| **DESCRIPTION** | Calculate the harmful effect of a waste |
| **INPUTS** | Depends on the product |
| **OUTPUTS** | None. |

|  |  |
| --- | --- |
| **NAME** | FR8 |
| **DESCRIPTION** | Determine if a biodegradable or recyclable waste is usable. |
| **INPUTS** | Depends on the type of waste |
| **OUTPUTS** | None. |

|  |  |
| --- | --- |
| **NAME** | FR9 |
| **DESCRIPTION** | List the waste of a product taking into account its harmful effects, showing first the most harmful. |
| **INPUTS** | Depends on the type of the wastes of the product |
| **OUTPUTS** | None. |

*Table 1 Functional Requirements*

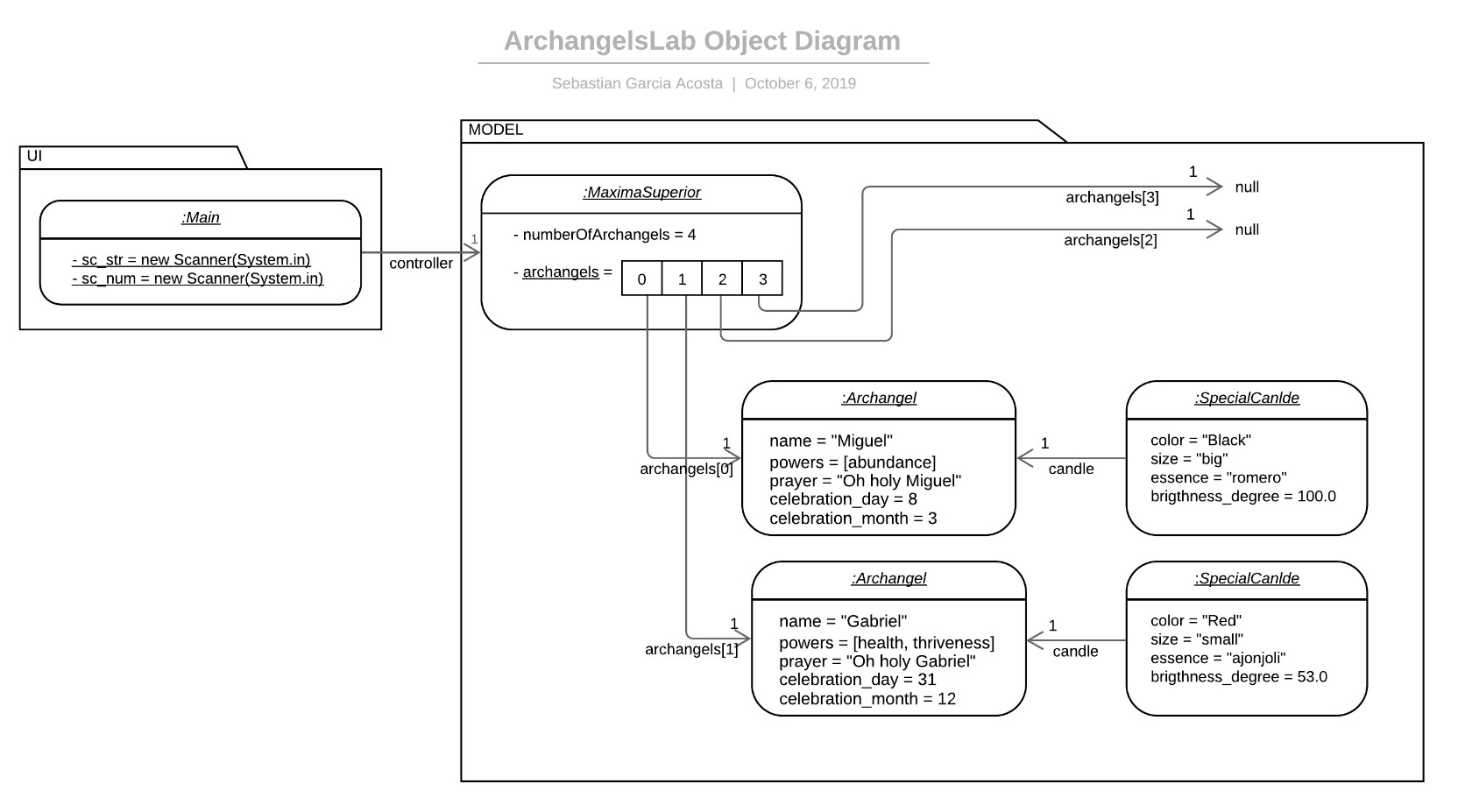
**UML Class Diagram**

****

You can find this UML Class Diagram on lucidchart:

<https://www.lucidchart.com/invitations/accept/06bfcad1-247e-4908-86e0-1db9629a2bb7>

# **Object diagram**

****

You can find this UML Object Diagram on lucidchart:

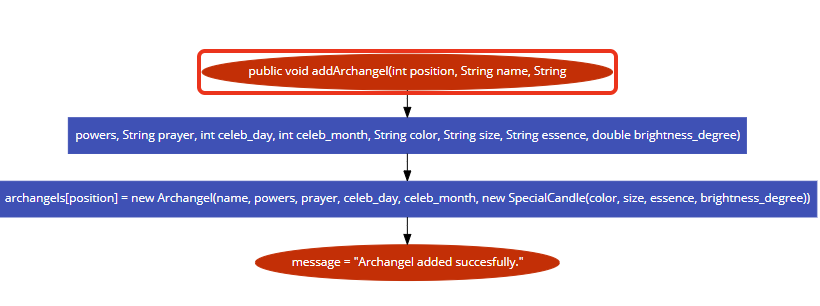
<https://www.lucidchart.com/invitations/accept/10d22738-d62a-4f5f-8528-0f363a2620c4>

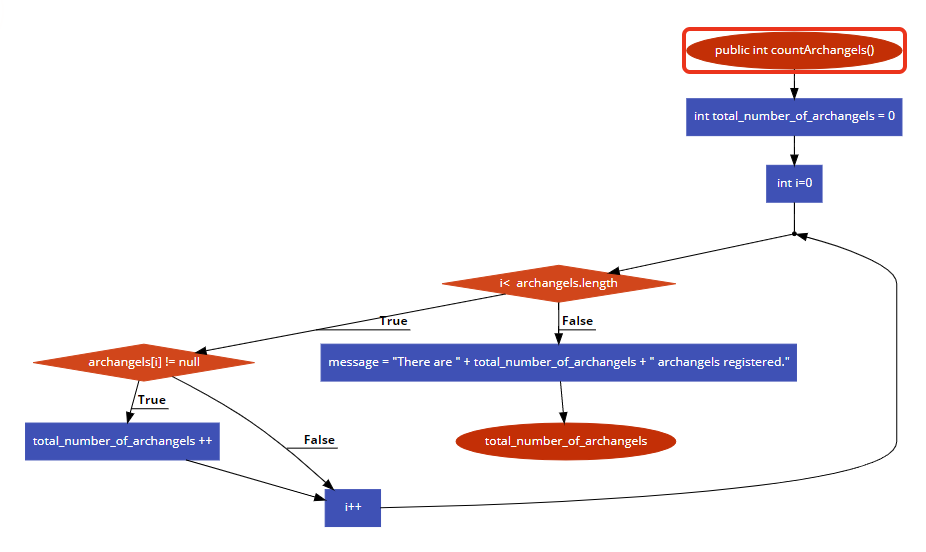
# **Traceability**

|  |  |  |
| --- | --- | --- |
| FUNCTIONAL REQUIREMENT | CLASS | METHOD |
| FR1 | Main  Main MaximaSuperior  MaximaSuperior  MaximaSuperior  Main  MaximaSuperior  MaximaSuperior  Archangel  MaximaSuperior  MaximaSuperior      MaximaSuperior MaximaSuperior | requestInfoAndAddArchangel()  requestName(String name)  nameIsValid(String name)  nameAlreadyExists(String name)  searchIdxByName(String name)  requestPowers()  powersAreValid(String[] powers)  searchByPowers(String[] powers)  searchPower(String power) RegistrationIsAllowed(String name, String[] powers, int celebration\_day, int celebration\_month)  addArchangel(int position, String name, String[] powers, String prayer, int celeb\_day, int celeb\_month, String color, String size, String essence, double brightness\_degree)  getNumArchangels() numOfAvailablePositions() |
| FR2 | MaximaSuperior | countArchangels() |
| FR3 | Main MaximaSuperior  Archangel Archangel | findByName() searchIdxByName(String name) getArchangels() toString() |
| FR4 | Main MaximaSuperior  Archangel Archangel | findByPowers() searchByPowers(String[] powers)  searchPower(String power) toString() |
| FR5 | Main MaximaSuperior  Archangel Archangel  Archangel Archangel SpecialCandle SpecialCandle | findByMonth(int month) searchCelebsByMonth(int month) getCelebrationMonth() getName()  getCelebrationDay() getCandle() getColor() getEssence() |
| FR6 | MaximaSuperior MaximaSuperior Archangel  Archangel Archangel MaximaSuperior | showAllCelebs() countArchangels() getName() getCelebrationDay() getCelebrationMonth() date2String(int day, int month) |
| FR7 | Main MaximaSuperior  MaximaSuperior  Main MaximaSuperior  MaximaSuperior  Archangel  MaximaSuperior  MaximaSuperior  Archangel Archangel | createPowers() nameAlreadyExists(String name)  searchIdxByName(String name) requestPowers()  powersAreValid(String[] powers)  searchByPowers(String[] powers)  searchPower(String power) appendPowers(String name, String[] powers) searchIdxByName(String name) getPowers() setPowers() |

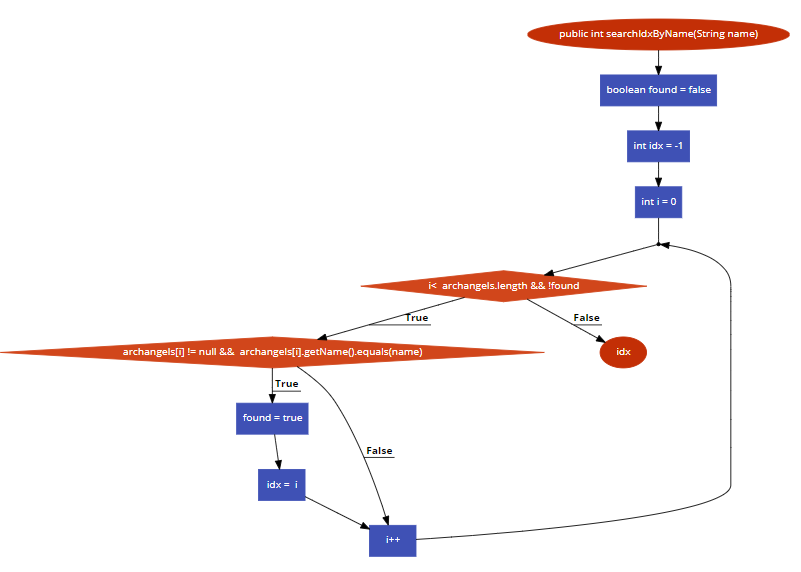
# **Flowcharts**

## FR1

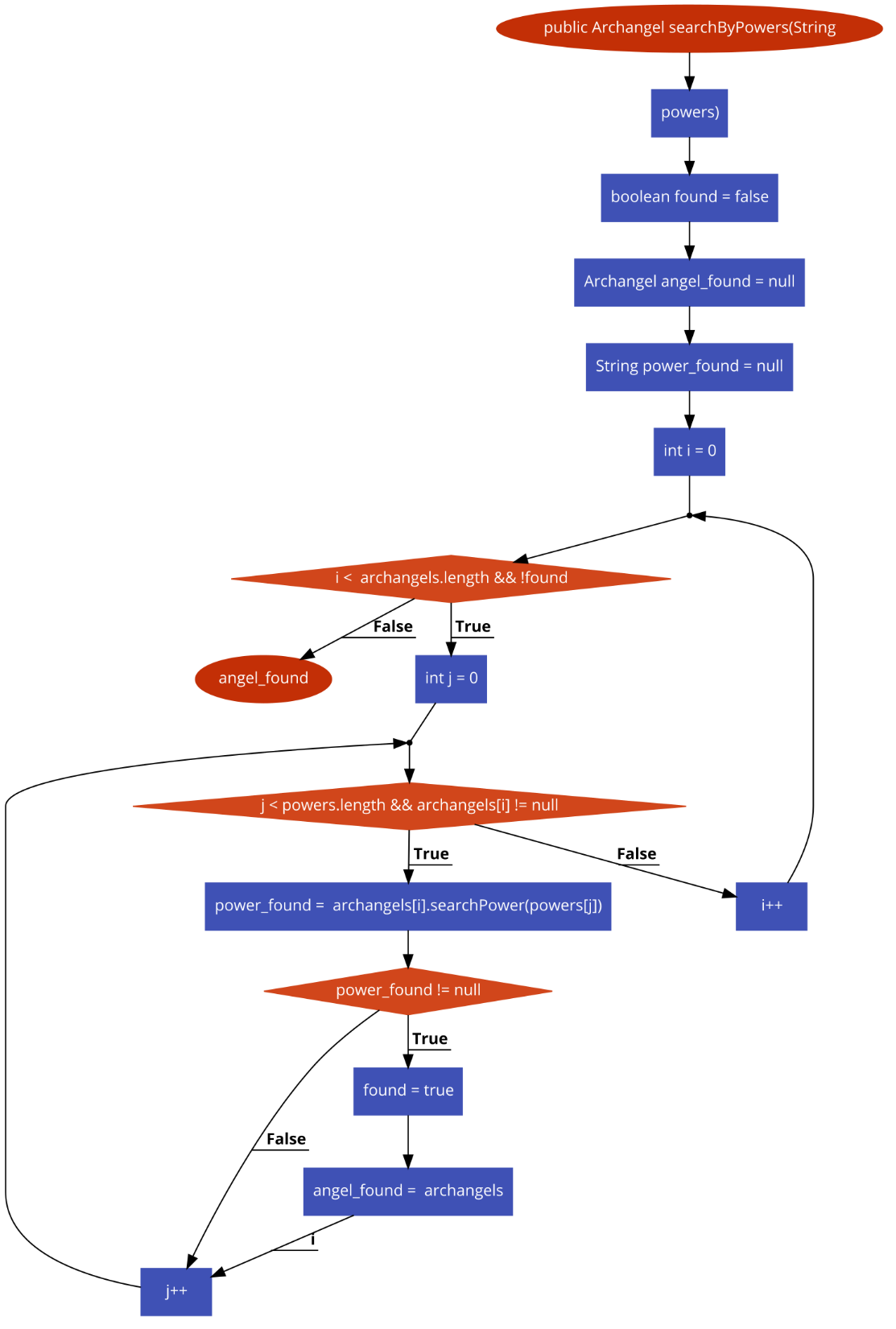
 FR2



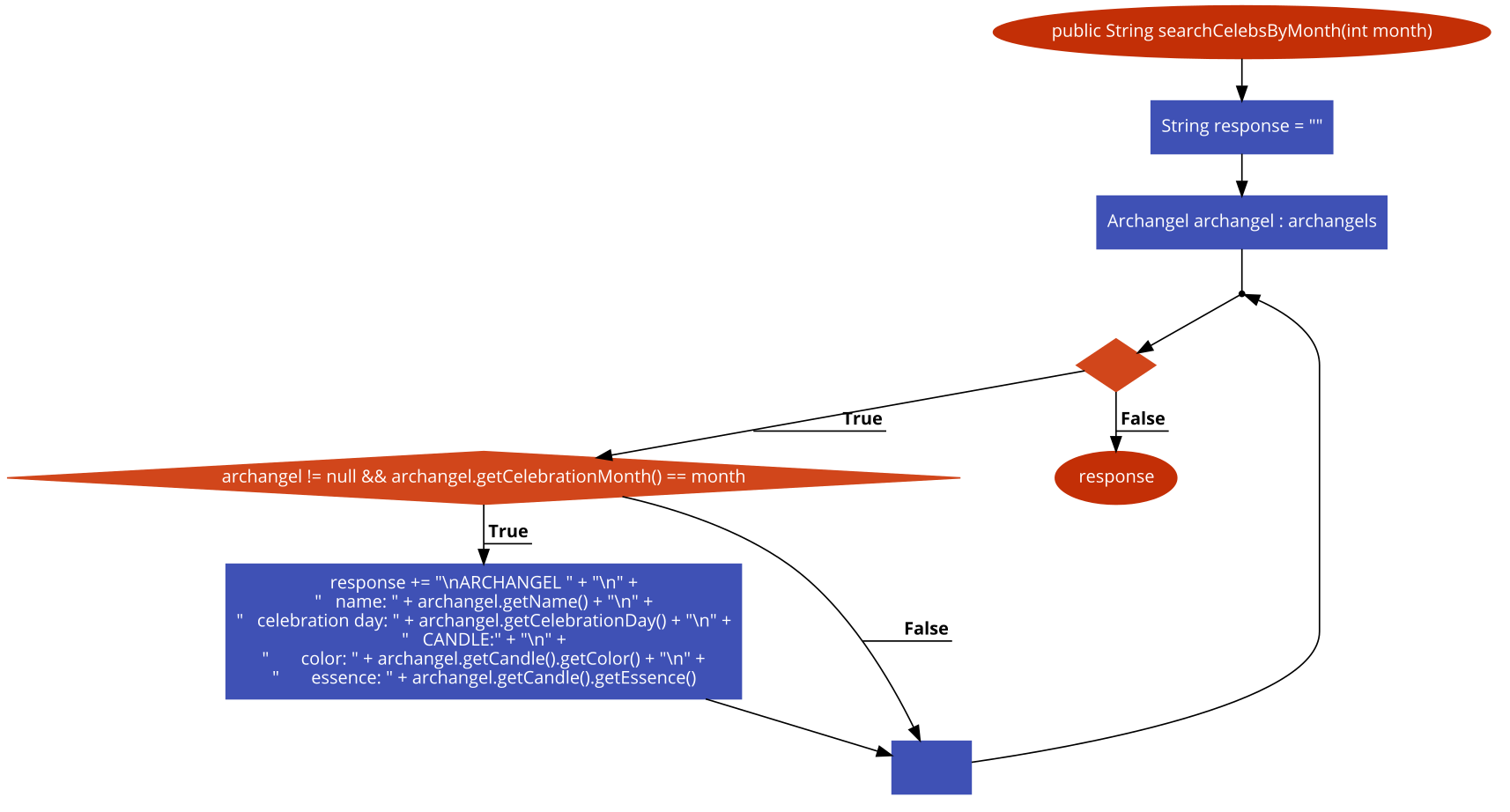
## FR3

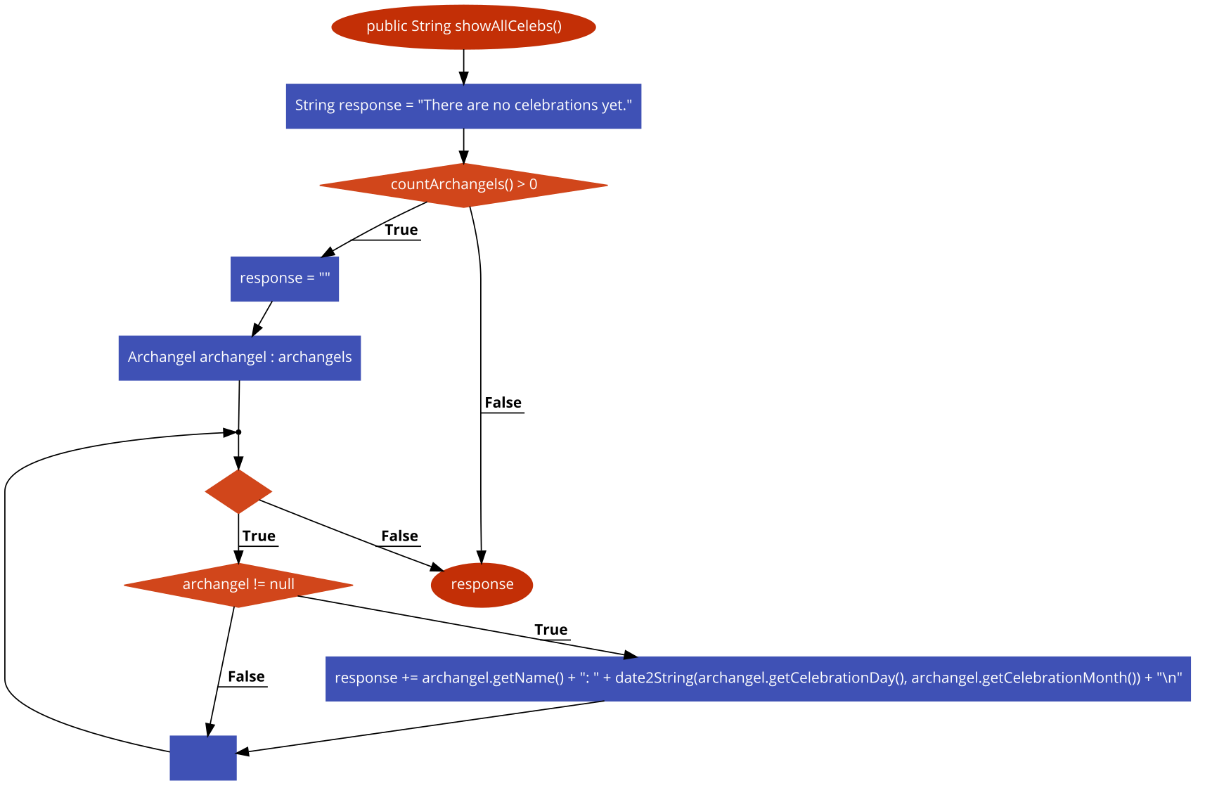


## FR4



## FR5

   
FR6



## FR7

