ФГАОУ ВО «Крымский федеральный университет имени В.И. Вернадского»

Физико-технический институт (структурное подразделение)

Кафедра компьютерной инженерии и моделирования

**Методология объектно-ориентированного моделирования с использованием UML**

(отчет по лабораторному практику

по дисциплине «Компьютерное моделирование»)

Выполнил:

студент 4 курса

группы ПИ-б-о-182(2)

Змитрович Никита Сергеевич

Симферополь 2021

**ВВЕДЕНИЕ**

Целью проекта является приобретение навыков по созданию компьютерных моделей и формализованных требований к информационным проектам.

Задачей проекта является формирование навыков применения:

* языка UML
* правил формирования требований
* принципов проектирования программных средств
* стандартов по оформлению программных документов

**Обзор проектируемой информационной системы:**

Информационная система представляет собой платформу “Мобильное приложение для метеостанции”, которая имеет клиент-серверную архитектуру. Она включает такие возможности, как просмотр температуры, влажности, давления, солнечного излучения, скорости и направления ветра.

**АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОЕКТИРОВАНИЯ**

1. Входные, выходные и внутренние данные процесса, включающие документы, сведения, информационные и управляющие воздействия, которые поступают в процесс, формируются в процессе или передаются из процесса в другие процессы.

|  |  |
| --- | --- |
| **Тип данных** | **Перечень данных** |
| Входные | Данные с метеостанции, предварительно обработанные сервером |
| Выходные | Графики и информация о текущей метео-ситуации. |
| Внутренние | Доменные данные, а также их не доменные аналоги (сетевые и т.д). |

1. Нормативные документы, устанавливающие требования к процессу:

Федеральный закон от 27.07.2006 N 152-ФЗ (ред. от 02.07.2021) "О персональных данных";

Федеральный закон от 29.12.2012 № 273-ФЗ “Об образовании в Российской Федерации”;

1. Участники процесса, структура их подчинённости и описание основных функций, представляющие собой схему подчинённости и перечень функций участников:

Рис. 1 Схема подчиненности

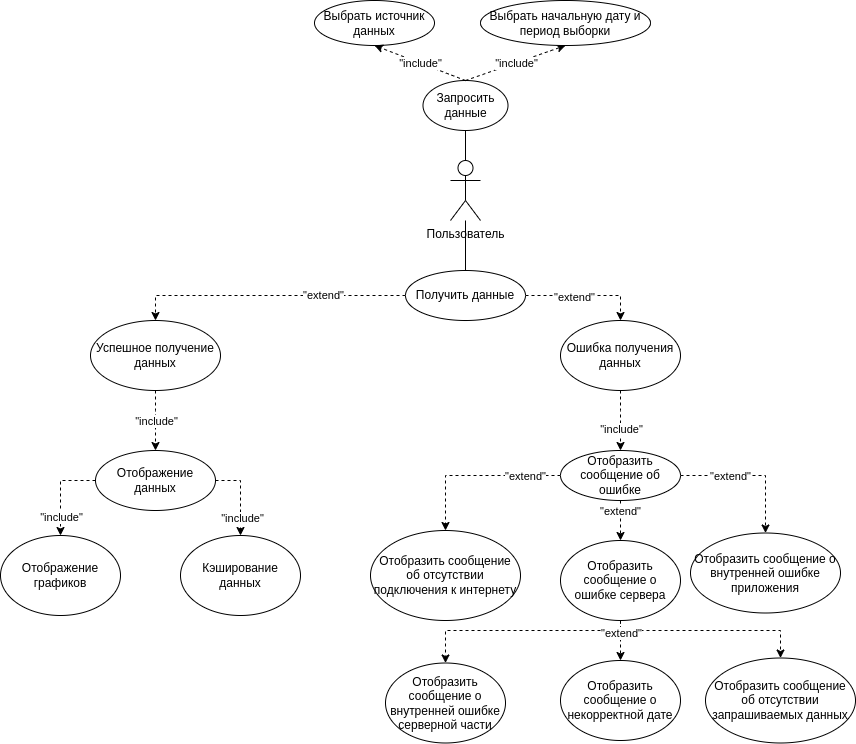
Пользователь выполняет следующие функции:

1. просмотр данных и графиков
2. изменение требований для данных
3. смена источника данных
4. обновление закешированных данных
5. Формирование пирамиды требований, содержащей уровень потребностей и уровень функциональных особенностей проектируемой системы

Таблица 1 - Формирование пирамиды требований

|  |  |
| --- | --- |
| Потребность | Функциональные особенности |
| Возможность выбора периода выборки данных | 1) Создание удобного и функционального интерфейса; |
| 2) Интерполяция данных в соответствии с периодом выборки; |
| Возможность просмотра последних запрошенных данных без наличия подключения к интернет сети | 1) Создание механизма кеширования данных; |
| 2) Создание соотвествующей структуры данных |
| Дальнейшая возможность расширения приложения | 1) Создание соотвествующей архитектуры |
| 2) Применение расшияремых подходов |
| 3) Написание документации |

1. Графическое описание функционального назначения системы (диаграмма вариантов использования).

Рис. 2 - Диаграмма вариантов использования

|  |
| --- |
| Authorisation |
| (-) \_login string  (+) Login string  (+) Password string  (-) \_sendAuthorisationRequest RelayCommand  (-) \_toRegistrationWindow RelayCommand  (-) \_toRecoveryWindow RelayCommand |
| (+) SendAuthorisationRequest() RelayCommand  (+) ToRegistrationWindow() RelayCommand  (+) ToRecoveryWindow() RelayCommand |

|  |
| --- |
| Recovery |
| (+)Email string  (-)\_sendRecoveryRequest RelayCommand  (-) \_\_toAuthorisationWindow RelayCommand |
| (+)SendRecoveryRequest() RelayCommand  (+)ToAuthorisationWindow() RelayCommand |

|  |
| --- |
| Registration |
| (-)\_login string  (-)\_\_email string  (-) \_\_\_phonenumber string  (+)PasswordConfirm string  (+)Login string  (+)Password string  (+)Email string  (+)PhoneNumber string  (-) \_sendRegistrationRequest RelayCommand  (-) \_toAuthorisationWindow RelayCommand |
| (+) SendRegistrationRequest() RelayCommand  (+) ToAuthorisationWindow() RelayCommand |

|  |
| --- |
| CreateCampaign  Контроллер всех campaing |
| (+) ExamForms List<ExamForm>  (+) Contests List<Contest>  (-) FormsOfEducation List<FormOfEducation>  (+) EducationLevels List<EducationLevel>  (+) Departments List<Department>  (+) Exams List<Exam>  (+) SpecialtiesToServer List<Specialty> |

|  |
| --- |
| CreateCampaignDepartmentPage |
| (-) \_page Page; (-) panelHeight int; (-) orderLabelId string;  (-) nameTextBoxId string; (-) codeTextBoxId string  (-) departmentGridId string; (-) departmentStackPanelId string  (-) departments List<Department>; (-) value int; (-) panelsId List<int>  (-) \_stackPanel StackPanel; (-) StackPanel StackPanel; (-) \_toMainWindow RelayCommand  (-) \_addDepartament RelayCommand; (-) \_deleteDepartament RelayCommand;  (-) \_test RelayCommand |
| (-) OnPreviewMouseDown() void ; (-) GotFocus() void  (+) ToMainWindow() RelayCommand; (+) AddDepartament() RelayCommand  (+) DeleteDepartament() RelayCommand; (+) Test() RelayCommand  (-) DeleteGrid() void; (-) CreateGrid() Grid  (-) GetDeleteButton() Button; (-) GetCodeTextBox() TextBox  (-) GetNameTextBox() TextBox; (-) GetOrderLabel() Label |

|  |
| --- |
| Exam |
| (+)Code int  (-)Name string  (+)ExamFormIds List<int> |
| (+) Exam() |