МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО

ОБРАЗОВАНИЯ

РУТ (МИИТ)

Кафедра «Автоматизированные системы управления»

**КУРСОВАЯ РАБОТА**

**По дисциплине «Технологии программирования»**

**По теме: «Проектирование и разработка информационной**

**системы завода»**

Группа: УВВ-211

Студенты:

Гущин Е.А., Сибилев И. А.,

Залунин А.В., Лопатников В..

Преподаватель: Давыдовский М.А.

Москва

2021 г.

**Задания по курсовой работе для дисциплины «Технологии**

**программирования»**

Задание выдается на группу из 2-4 человек и выполняется на лабораторных занятиях и

занятиях по курсовой работе.

Язык программирования - Python.

Описание проекта разрабатываемой информационной системы должно быть выполнено на

языке UML.в среде UML Designer.

Документация по заданию должна содержать:

1. Задание по курсовой работе.

2. Диаграммы на языке UML.

3. Краткое описание каждого класса и метода.

4. Руководство пользователя со скриншотами интерфейса программы.

5. Набор тестов и результатов для отладки отдельных методов, модулей и

комплексной отладки задания.

6. Тексты программ с комментариями

Диаграммы языка UML:

• Диаграмма вариантов использования (Use Case Diagram)

• Диаграмма классов (Class Diagam)

• Диаграмма деятельности (Activity Diagram)

• Диаграмма состояний (State Machine Diagram)

• Диаграмма последовательности (Sequence Diagram)

• Диаграмма компонентов (Component Diagram)

• Диаграмма развертывания (размещения) (Deployment Diagram)

Необходимо разработать информационную систему в соответствии с заданием. В задании

перечислены пользователи системы (актеры) и варианты использования системы

актерами. Можно скорректировать список актеров и вариантов использования по

согласованию с преподавателем.

Предлагаемый перечень заданий может быть дополнен любым другим заданием по

согласованию с преподавателем. Приветствуется разработка проекта на основе реальных

заданий.

Общие требования по выполнению задания:

• Каждый пользователь информационной системы должен регистрироваться в

системе сам или это делает администратор

• При входе в систему каждый пользователь вводит свой логин и пароль

• Данные в системе хранятся в файлах

• При запуске системы все данные из файлов считываются в оперативную память и

хранятся в форме таблиц.

• По завершению работы с системой данные из таблиц записываются обратно в

файлы. Файлы при этом не обновляются, а создаются заново на основании данных

из таблиц

1. **Интернет-магазин. Доставка товаров**

Администратор

* Составление списка товаров

Кладовщик

* Формирование заказа (подбор товаров по заказу пользователя)

Диспетчер

* Формирование набора заказов для доставки курьером
* Слежение за доставкой заказов

Курьер

* Доставка заказов. После доставки заказа информация сразу поступает к диспетчеру

**Распределение задач:**

Гущин Е.А.:

• Построение UML-диаграмм;

• Создание графического интерфейса приложения;

• Написание классов-контроллеров к графическому интерфейсу;

Сибилев И. А.:

• Построение UML-диаграмм;

• Написание тестов к методам классов;

• Руководство пользователя;

Залунин А.В.:

• Построение UML-диаграмм;

• Написание классов-списков;

• Работа с базами данных;

Лопатников В..:

• Построение UML-диаграмм;

• Написание классов-сущностей;

• Тестирование работы программы;

**Компонент “администратор ”**

|  |  |
| --- | --- |
| Поток “Регистрация нового работница” | |
| Действие исполнителя | Отклик системы |
| 1. Заполнение информации о  работнике, ввод логина и пароля, выбор должности |  |
| 2. Подтверждение | 1. Проверяет, нет данного логина в базе |
|  | 4.1 Записывает работника в базу данных |
|  | 4.2 Просит проверить правильность введённых данных |

|  |  |
| --- | --- |
| Поток “Удаление работника” | |
| Действие исполнителя | Отклик системы |
|  | 1. Выводит список сотрудников |
| 1. Выбор сотрудника для удаления |  |
| 3. Подтверждение | 4. Удаляет выбранного сотрудника из базы данных |

|  |  |
| --- | --- |
| Поток “Добавление новой категории товаров” | |
| Действие исполнителя | Отклик системы |
| 1. Заполнение информации о новом товаре | 2. Проверяет на наличие повторов |
|  | 3.1 Заносит новый товар в базы данных |
|  | 3.2 Просит проверить введённые данные |

|  |  |
| --- | --- |
| Поток “Удаление категории товаров” | |
| Действие исполнителя | Отклик системы |
|  | 1. Выводит список товаров |
| 2. Выбор товара из списка | 3. Удаляет категорию |

**Компонент “кладовщик”**

|  |  |
| --- | --- |
| Поток “Сборка заказа” | |
| Действие исполнителя | Отклик системы |
| 1. Выбор заказа для сборки | 2. Удаление нужного количества товаров из базы данных склада |
|  | 3. Добавление нового заказа в список |

|  |  |
| --- | --- |
| Поток “Удаление заказа” | |
| Действие исполнителя | Отклик системы |
|  | 1. Показывает список заказов |
| 2. Выбор заказа для отмены |  |
| 3. Подтверждение | 4. Добавление нужного количества товаров в базу данных склада |
|  | 5. Удаление заказа из списка |

|  |  |
| --- | --- |
| Поток “Списание товаров” | |
| Действие исполнителя | Отклик системы |
| 1. Ввод id товара и его количества | 1. Удаляет нужное количество товаров из базы данных |

|  |  |
| --- | --- |
| Поток “Учёт поступлений товара” | |
| Действие исполнителя | Отклик системы |
| 1. Ввод id товаров и их количество | 1. Добавляет нужное количество товаров в базы данных |

|  |  |
| --- | --- |
| Поток “Обработка возврата” | |
| Действие исполнителя | Отклик системы |
|  | 1. Выводит сообщение о возврате |
| 2. Подтверждение возврата | 3. Добавляет нужное количество товаров в базы данных |

**Компонент “Диспетчер”**

|  |  |
| --- | --- |
| Поток “Формирование наборов” | |
| Действие исполнителя | Отклик системы |
|  | 1. Загружает все заказы ожидающие доставки из базы данных |
| 1. Формирование набора | 3. Добавление нового набора в список ожидающих доставки |

**Компонент “Курьер”**

|  |  |
| --- | --- |
| Поток “Доставка заказов” | |
| Действие исполнителя | Отклик системы |
|  | 1. Выводит информацию о заказах и адресах доставки |
| 2. Взятие набора заказов в доставку | 1. Изменяет статус заказа, сообщает диспетчеру |
| 4.1 Подтверждение доставки заказа |  |
| 4.2 Возникла проблема |  |

|  |  |
| --- | --- |
| Под поток “Подтверждение доставки заказа” 4.1 | |
| Действие исполнителя | Отклик системы |
| 1. Докладывает об успешной доставке | 2. Изменяет статус заказа, сообщает диспетчеру |

|  |  |
| --- | --- |
| Под поток “Возникла проблема” 4.2 | |
| Действие исполнителя | Отклик системы |
| 1. Сообщить об ошибке | 2. Изменяет статус заказа, сообщает диспетчеру |

**Диаграммы на языке UML**

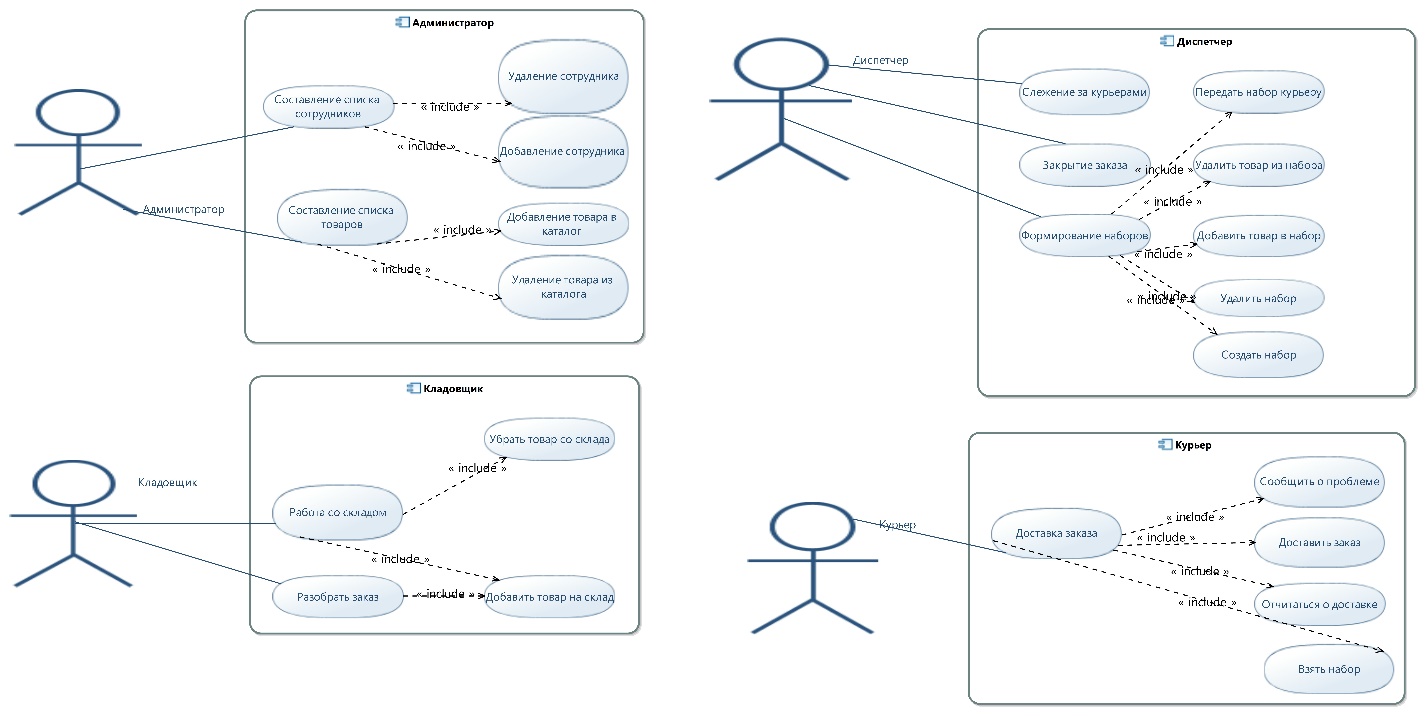
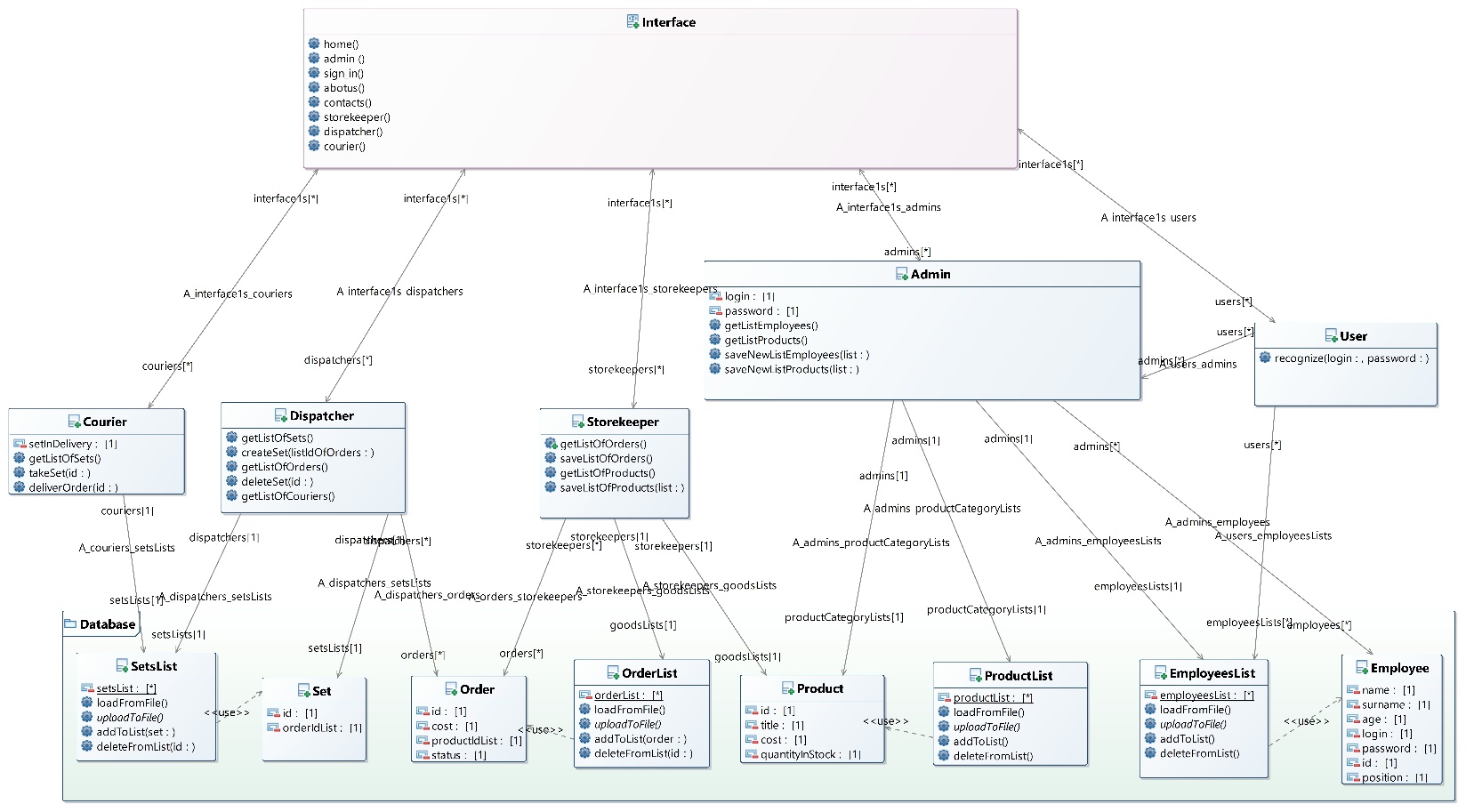
Диаграмма вариантов использования (Use Case Diagram):****

Диаграмма классов(Class Digram):



1. Класс Employee - класс сущность, хранящая данные о работнике

1.1 Поле Id - хранит id работника

1.2 Поле name - хранит имя работника

1.2 Поле surname - хранит фамилию работника

1.3 Поле age - хранит возраст работника

1.4 Поле login - хранит логин работника

1.5 Поле password - хранит пароль работника

1.6 Поле position - хранит должность работника

2. Класс Good - класс сущность, хранящая данные о товаре

2.1 Поле id - хранит id товара

2.2 Поле cost - хранит цену товара

2.3 Поле title - хранит название товара

2.4 Поле quantityInStock - хранит количество товара на складе

3. Класс Order - класс сущность, хранящая данные о заказе

3.1 Поле id - хранит id заказа

3.2 Поле goods\_id\_list - хранит id товаров в составе заказа

3.3 Поле status - хранит стату заказа

4. Класс OrdersSet - класс сущность, хранящая данные о наборе заказов

4.1 Поле id - хранит id набора

4.2 Поле orders\_id\_list - хранит id заказов в составе набора

4.3 Поле status -хранит статус набора

5. Класс EmployeesList - класс для управления базой данных

5.1 Поле \_\_List - хранит загруженный список работников в формате классов Employee

5.2 Метод download\_from\_file - выгружает данные из базы данных и загружает в \_\_List

5.3 Метод download\_to\_file - загружает данные в базу данных из \_\_List

5.4 Метод add\_new\_worker - добавляет нового работника в \_\_List

5.5 Метод delete\_worker - удаляет работника из \_\_List по id

5.6 Метод get\_list - возвращает \_\_List

6. Класс GoodsList - класс для управления базой данных

6.1 Поле \_\_List - хранит загруженный список работников в формате классов Good

6.2 Метод download\_from\_file - выгружает данные из базы данных и загружает в \_\_List

6.3 Метод download\_to\_file - загружает данные в базу данных из \_\_List

6.4 Метод add\_new\_good - добавляет новый товар в \_\_List

6.5 Метод delete\_good - удаляет товар из \_\_List по id

6.6 Метод get\_list -

7. Класс OrdersList - класс для управления базой данных

7.1 Поле \_\_List - хранит загруженный список работников в формате классов Order

7.2 Метод download\_from\_file - выгружает данные из базы данных и загружает в \_\_List

7.3 Метод download\_to\_file - загружает данные в базу данных из \_\_List

7.4 Метод add\_new\_order -добавляет новый заказ в \_\_List

7.5 Метод delete\_order - удаляет заказ из \_\_List по id

7.6 Метод get\_list -

8. Класс OrdersSetsList - класс для управления базой данных

8.1 Поле \_\_List - хранит загруженный список работников в формате классов OrderSet

8.2 Метод download\_from\_file - выгружает данные из базы данных и загружает в \_\_List

8.3 Метод download\_to\_file - загружает данные в базу данных из \_\_List

8.4 Метод add\_new\_ordersset - добавляет новый набор заказов в \_\_List

8.5 Метод delete\_ordersset - удаляет набор заказов из \_\_List по id

8.6 Метод get\_list -

9. Класс Courier - управляющий класс, обеспечивающий работу курьера

9.1 Поле id - хранит id курьера

9.2 Поле setInDelivery - Хранит id набора, который сейчас доставляет курьер

9.3 Метод getListOfSets - возвращает список наборов заказов

9.4 Метод takeSet - Указать выбранный набор как находящийся в доставке

9.5 Метод deliverOsrder - Указать выбранный набор как доставленный

1. Класс Dispatcher - управляющий класс, обеспечивающий работу диспетчера

10.1 Поле Id - хранит id диспетчера

10.2 Метод getListOfSets - возвращает список наборов заказов

10.3 Метод ctreateSet - записывает новый набор заказов в базу данных

10.4 Метод getListOfOrders - возвращает список заказов

10.5 Метод deleteSet - удаляет выбранный набор заказов по id

10.6 Метод getListOfCouriers - возвращает список курьеров

11. Класс Storekeeper - управляющий класс, обеспечивающий работу кладовщика

11.1 Поле Id - хранит id кладовщика

11.2 Метод getListOfOrders - возвращает список заказов

11.3 Метод saveListOfOrders - записывает новый список заказов в базу данных

11.4 Метод getListOfProducts -возвращает список товаров

11.5 Метод saveListOfProducts -записывает новый список товаров базу данных

12. Класс Admin

12.1 Поле login - хранит логин админа

12.2 Поле password - хранит логин админа

12.3 Метод getListEmployees - возвращает список работников

12.4 Метод getListProducts - возвращает список товаров

12.5 Метод saveNewListEmployees - записывает новый список работников в базу данных

12.6 Метод saveNewListProducts - записывает новый список товаров базу данных

13. Класс User - класс пользователя

13.1 Метод User - Возвращает работника подходящего

14. Класс Interface - класс интефейса

14.1 Метод home - возвращает страницу home.html

14.2 Метод admin - возвращает страницу admin.html

14.3 Метод sing\_in - возвращает страницу sing\_in.html

14.4 Метод abotus - возвращает страницу abotus.html

14.5 Метод contacts - возвращает страницу contacts.html

14.6 Метод storekeeper - возвращает страницу storekeeper.html

14.7 Метод dispatcher - возвращает страницу dispatcher.html

14.8 Метод courier - возвращает страницу courier.html

Изображение выглядит как стол

Автоматически созданное описание