|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| VILNIAUS KOLEGIJA  UNIVERSITY OF APPLIED SCIENCES  FACULTY OF ELECTRONICS AND INFORMATICS  Image result for viko logo | | |  | | | VILNIUS COLLEGE  Image result for viko logoFACULTY OF ELECTRONICS AND INFORMATICS |
|  | | |  | | |  |
| **INFORMATION SYSTEMS** | | |  | | | **INTRODUCTION TO INFORMATICS** |
| PRACTICAL ASSIGNMENT  The System Requirements Specification  6531BX028 PI18E | | |  | | | PRACTICAL ASSIGNMENT  SPOTIFY USER MANUAL  6531BX028 PI18E |
| STUDENT | DŽIUGAS PEČIULEVIČIUS | STUDENT | | DŽIUGAS PEČIULEVIČIUS |
| (SIGNATURE)  12/10/2019 | | |  | | | LECTURER |
| LECTURER | TATJANA LIOGIENĖ | (SIGNATURE)  10/17/2018 | | VIRGILIJUS KUKLIERIUS |
| (SIGNATURE)  12/10/2019 | | |  | | | 2018 |

2019

**TABLE OF CONTENTS**

[INTRODUCTION 4](#_Toc24977788)

[A. 5](#_Toc24977789)

**TABLE OF FIGURES**

No table of figures entries found.

# INTRODUCTION

**Program used for charts**

Website: draw.io

**Goal**

A goal is to prepare system Requirements specification document

**Tasks**

* Describe a problem from the second practical work and offer a solution to fix it.
* List information system users (at least two) and describe them.
* Create the list of functional requirements for new information system.
* Create the list of nonfunctional requirements.
* Create UML Use Case diagram at least 7 use cases, at least 1 <<include>>, at least 1 <<extend>>, actor generalization (correct PW2 errors).
* Describe each use case primary and secondary scenarios.
* Create UML Activity diagram. Requirements for Activity diagram: two swimlanes.
* Find at least 3 classes. Find attributes and methods/operations of classes.

# PROMBLEM AND A SOLUTION

In the second practical work the uncovered problem was that all the orders were not tracked by the system and everything had to be done by hand. There was no item tracking so the item could be lost easier. Also no messages have been sent to the managers if the items are running out in the storage and more need to be ordered.

# SYSTEM USERS

1. **Office department manager** – responsible for receiving orders and sending a plan to material warehouse on distributing items across manufacturing department.
2. **Raw-material warehouse** manager – receives plan and distributes material according to plan, if anything is missing – new items will be ordered by sending an order to the office department.
3. **Manufacturing department manager** –responsible for assembling, packaging, testing and labeling of the computers. Receiving material and ordering parts that are missing.
4. **Product warehouse manager** – Responsible for receiving all working items and storing them.

# LIST OF FUNCTIONAL REQUIREMENTS

1. System should let every manager log-in into and log-out of their own accounts.
2. The system should have functionality to let the office manager to send a full plan with item numbers, item identification numbers.
3. The system should let leave a note for any specific requests to the Raw-material warehouse manager and Product Warehouse manager.
4. The system should let departments to receive plans and material information that are specified above when they arrive and confirm the goods
5. Should let department managers to place orders of new items if needed and send the request to the office.
6. The system should check storage and send an alert to a manager when items are running low and more of them need to be ordered.

# LIST OF *NON-FUNCTIONAL* REQUIREMENTS

1. **Operational**

* The system should run on PCs that are used by one of the managers specified.
* The system should interface with the material tracking system.
* The system should have an app dedicated to this system.

1. **Performance**

* The system should be available all year long any time of the day.
* The system should update storage capacity for the system every day at 5PM.

1. **Security**

* Only managers specified can access this system.
* Managers can check the list of items in the inventory.
* Managers can check tracking numbers and where the item is.
* Managers can receive and send plans.
* The system should be protected from attacks and other kind of viruses.

1. **Cultural and Political**

* The system should be able to distinguish between Europe currency and any other currency from other nations.

# UML USE CASE DIAGRAM

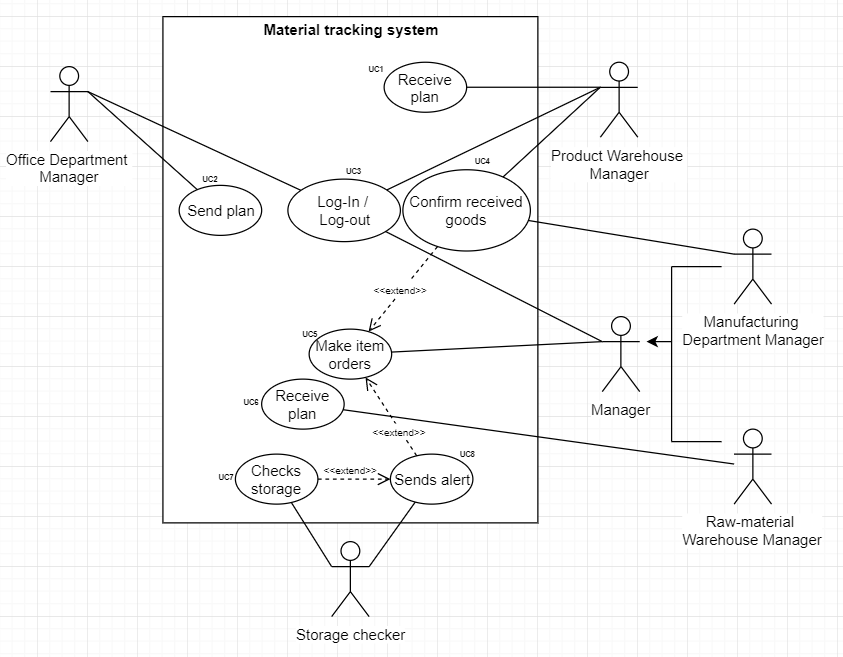


Figure 1 - use case diagram

# PRIMARY AND SECONDARY SCENARIOS

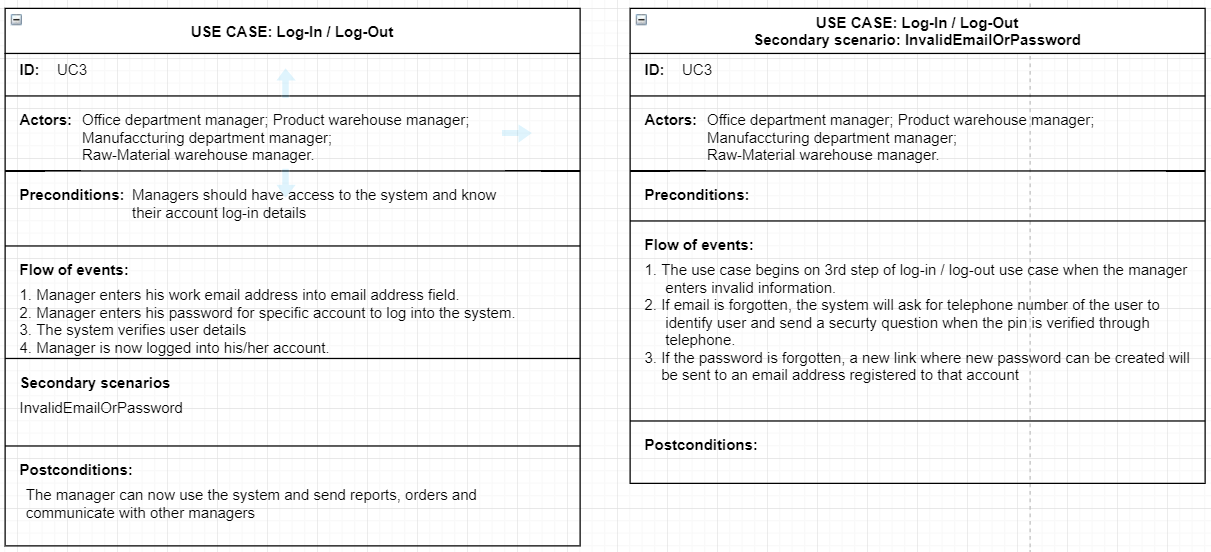


Figure 3 - log-in / log-out use case

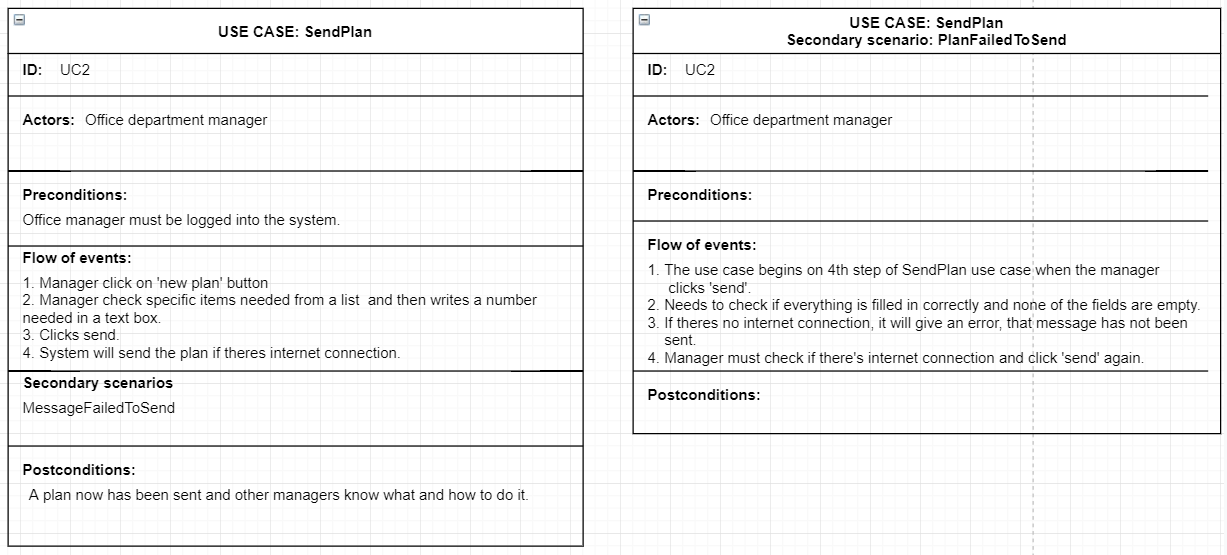


Figure 4 – Send Plan use case

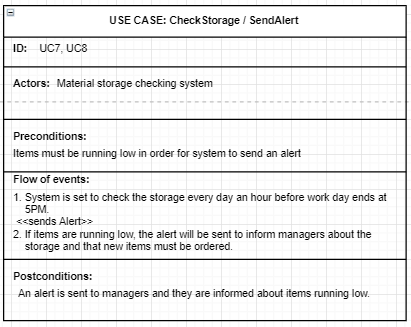


Figure 5 - Check Storage / Send Alert use case

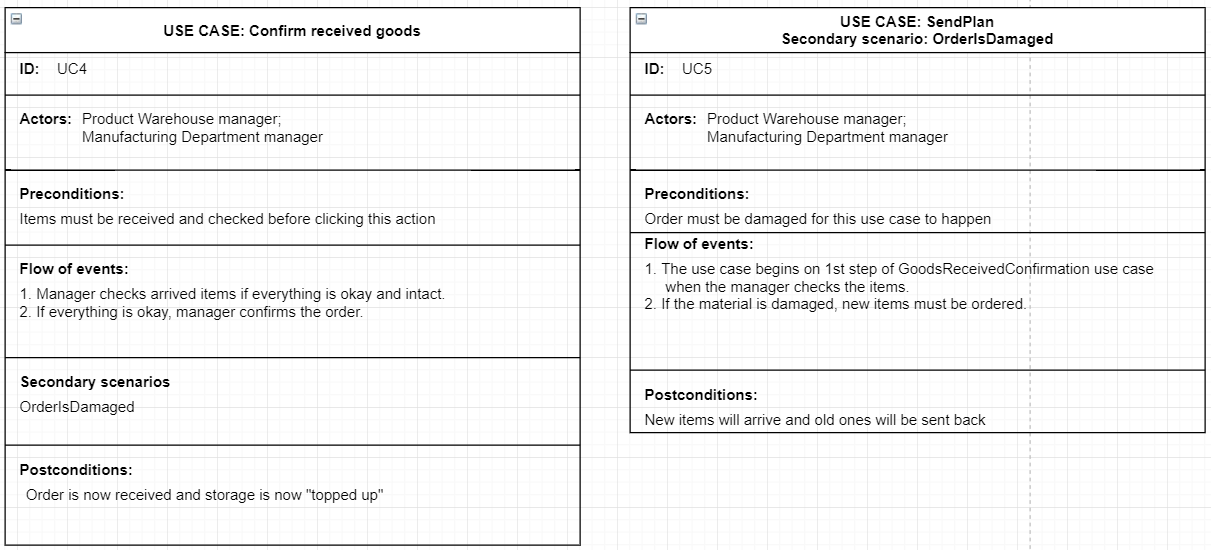


Figure 6 – Good Received Confirmation use case

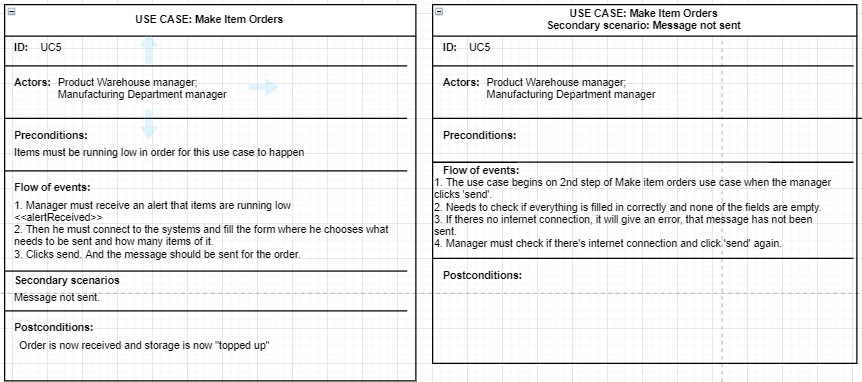
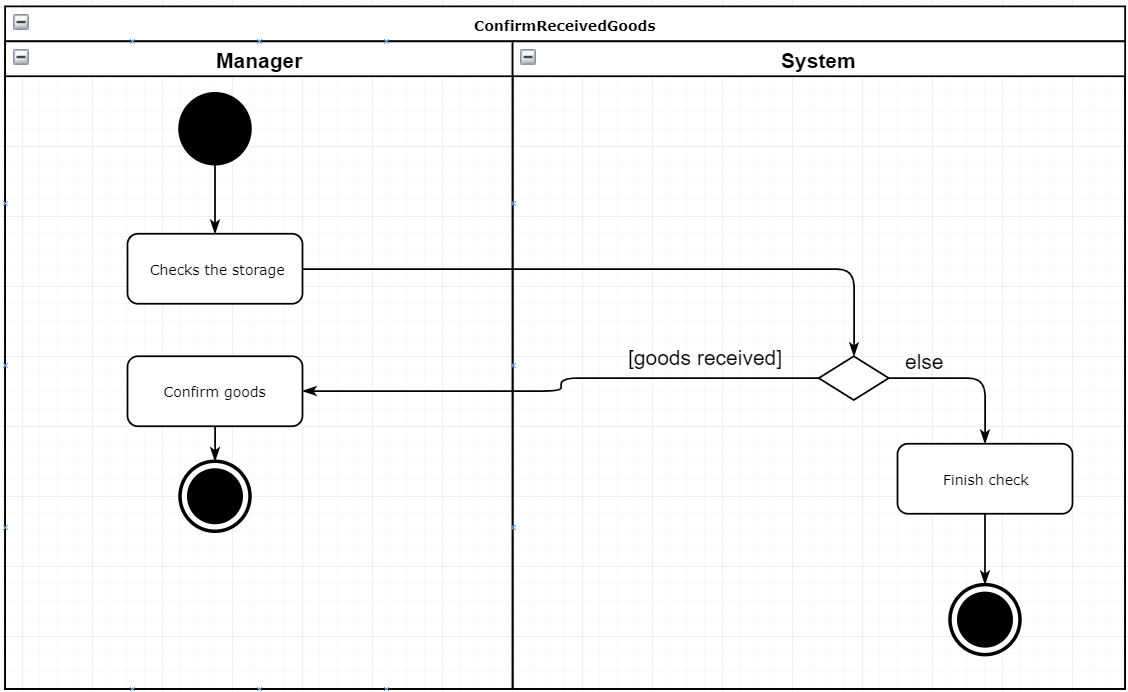
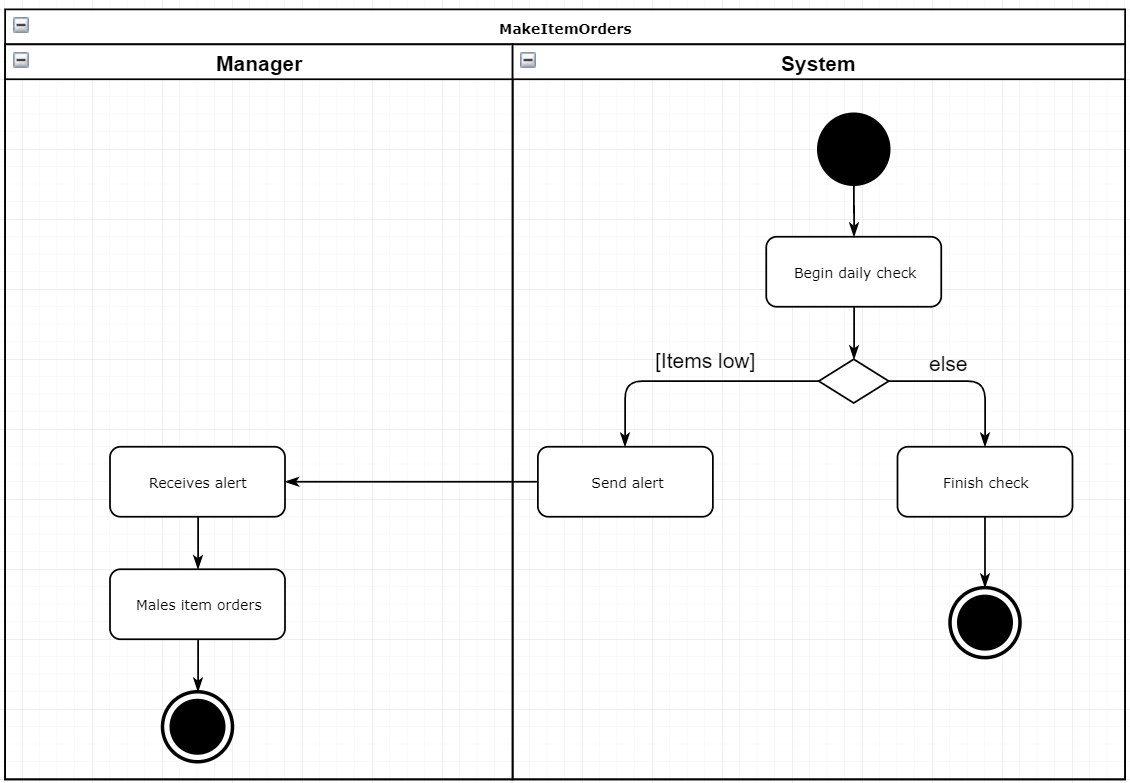
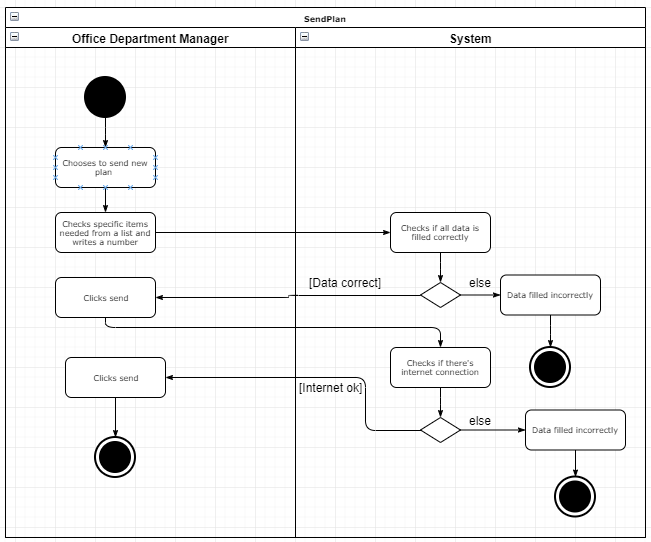
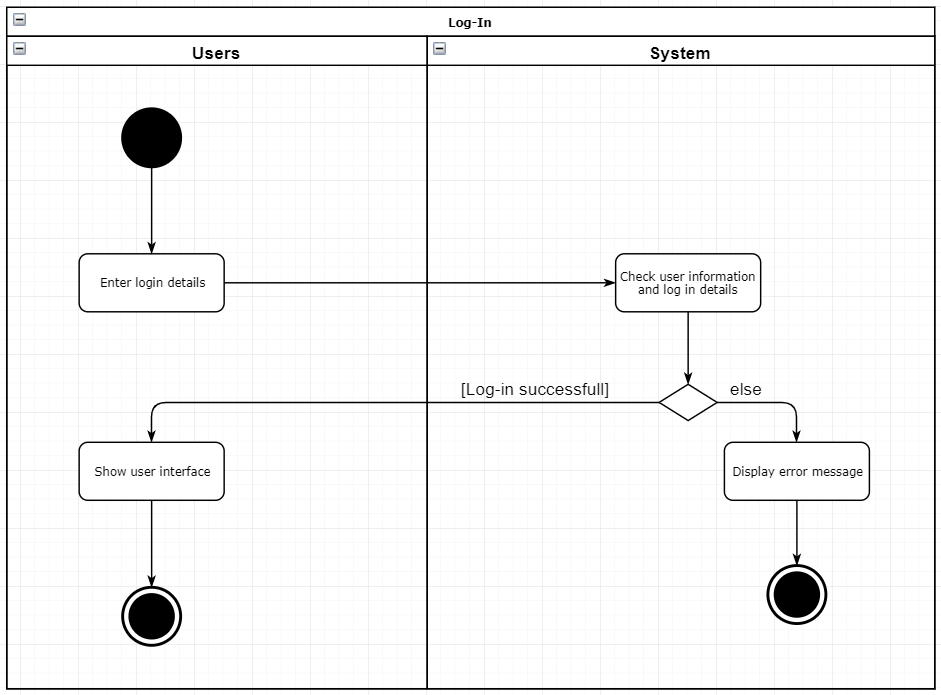


Figure - Make item orders

# UML ACTIVITY DIAGRAM



# CLASSES