**List of requirements No. 2:**

1.Formation of a list of requirements for the project. Specification and validation of requirements. Use-case model.

* Creation of initial project documentation in the selected language (Ukrainian, English).
* Availability of the project name and description.
* Availability of a table or list of system roles with a description of each of them.
* Identification of system subsystems.
* Availability of a table with use cases and their brief descriptions for each subsystem and role.
* Availability of use-case diagrams corresponding to the above table using UML.
* Correctness of the specified roles, subsystems and corresponding use cases.
* Completeness of the system in the sense of having all the main use cases.
* Correct formatting of the document.
* Availability of a report on the work performed.

**My part:**

* Created a table/list of system roles with a description of each of them:

**Role table:**

|  |  |  |
| --- | --- | --- |
| № | Role | Description |
| 1 | Guest | A user that isn’t logged in. Can register/log in or analyze data from self-provided csv file. Can also use guest log in that allows to view but not interact with various functionality and read ToS of relevant subsystems. |
| 2 | Monoposter | Can use functionality of all subsystems, except User Management System: manage account information, request data extraction/deletion, interact with monobank api to track progress of jar(s), use data science, connect social media apps to post to media and/or text to various social media apps at once. |
| 3 | Admin | Authorized user with elevated privileges. Can use their own subsystem to view system logs, user information and manage various user information. |

* Identification of system subsystems:

**Subsystems:**

|  |  |  |
| --- | --- | --- |
| № | Subsystem name | Available functionality description |
| 1 | Authorization System | Allows users to sign up, log in, log out and reset a password. Guest users can log in using “guest login”. |
| 2 | Account Management System | Allows users to change their account information (i.e. first and last name), request partial/full data extraction/deletion. |
| 3 | Monobank Management System | Connect/disconnect monobank app through changing login credentials, connect/disconnect jars, refresh data of connected jar(s). Upload csv file(s) for further analysis, display information from chosen jar(s)/file(s) as a table, analyze data from chosen jar(s)/file(s), save acquired data to file/server. |
| 4 | Posting System | Connect/disconnect social media apps through updating relevant credentials. Enter text(captions) and/or upload relevant media file(s) and save (part of) this information as a template. Manage templates: choose and use/update relevant template/text and/or media for posting, choose social media apps to post it to, view engagement stats of last few posts. |
| 5 | User Management System | Add/view/update/delete information about individual user(s), such as their login information, template(s), social media and monobank credentials, view system logs, time out/ban users or their usage of particular credentials/apps when they violate ToS. |

* Formatted the final document

**Link to my part:**

https://docs.google.com/document/d/179x-x1k7a0u935fJFISNKtNWwSSVfldBrxhpSZo7NEM/edit

**Link to the entire document:** https://docs.google.com/document/d/1CmGX\_5DZN1xm\_ofgaJm0zgvEXesJm7sZwRd58ep2ryU/edit