Technical Report

https://github.com/5c0rp264/C-Project



November 24, 2020

GROUP 3 - SN0FX221

CONFIDENTIALITY SHEET OF THE REPORTS CESI – SCHOOL OF ENGINEERS

Title of the report: EasySave, the revolution of backup softwares

Team members: LHINARES TANGUY | AOUSTIN QUENTIN | VINCENT JACQUES | JÉRÉMY

GABRIEL | RÉMI MOUNIER

Report due date : 23/11/2020

Confidentiality of the report:

The reports/memoirs are kept in an archive and can be freely consulted. They can be used by the recipients, the studies can be published ...

 \square Restricted circulation

The reports / dissertations are returned **to the students** at the end of the defense. No reproduction is authorized. Responsibility for this operation is entrusted to the students. Within the framework of the policy against plagiarism, the reports / dissertations will be likely to be analyzed to verify their sources, regardless of the mode of distribution provided for above.

Acknowledgements

We would like to thank all those who contributed to the full realization of this report and to the various opinions provided during its development, as well as for their help in understanding the various tools used (Azure DevOps, Pipelines on Github...).

In particular, we would like to thank our professor, **ANNE LAURE GAUDON of CESI – SAINT-NAZAIRE** who guided us in the development of the deliverable. Her listening and advice during our exchanges by email allowed us to complete this part of the project within the time allowed but also to develop our personal reasoning and to better understand the whole subject.

Abstract

Our team has just integrated the software publisher ProSoft. Under the responsibility of the CIO, we are in charge of managing the "EasySave" project which consists in developing a backup software. As any software of the ProSoft Suite, the software will be integrated into the pricing policy.

- Unit price: 200 €HT
- Annual maintenance contract 5/7 8-17h (updates included): 12% purchase price (Annual contract tacitly renewed with revaluation based on the SYNTEC index)

During this project, we will have to ensure the development, the management of major and minor versions, but also the documentation (user and customer support). To ensure that our work can be taken over by other teams, we must work within certain constraints such as the tools used (see <u>Software restrictions</u>).

Release dates and versions:

Version 1.0: 25/11/2020
Version 2.0: 07/12/2020
Version 3.0: 17/12/2020

Table Of Contents

CONFIDENTIALITY SHEET OF THE REPORTS CEST—SCHOOL OF ENGINEERS	
ACKNOWLEDGEMENTS	2
ABSTRACT	
TABLE OF CONTENTS	4
INTRODUCTION	5
PROJECT MANAGEMENT	6
EASYSAVE AND DOCUMENTATION	9
LOG AND STATE FILES	15
Unified Modeling Language - Diagrams	17
CONCLUSION	21
Personal project review :	21
APENDIXES	22
Annex 1 : CODE_OF_CONDUCT.MD	22
Annex 2 : CONTRIBUTING.MD	23
Annex 3 : LICENSE	24
Annex 4: PULL_REQUEST_TEMPLATE.MD	25
Annex 5 : SECURITY.MD	26
Annex 6 : bug_report.md	27
Annex 7 : feature_request.md	28
Annex 8 : Worflows -> codegl-analysis.yml	

Introduction

This document aims to present the first version of the EasySave the features are as follows:

The software is a Console application using .Net Core. It must allow the creation of up to 5 backup jobs.

A backup job is defined by:

- An appellation
- A source directory
- A target directory
- One type
- Full
- Differential
- English

The user may request the execution of one of the backup jobs or the sequential execution of the iobs.

The directories (sources and targets) can be on local, external or network drives. All the elements of the source directory are concerned by the backup.

Daily Log File:

The software must write in real time in a daily log file the history of the actions of the backup jobs. The minimum expected information is:

- Timestamp
- Naming the backup job
- Full address of the Source file (UNC format)
- Full address of the destination file (UNC format)
- File Size
- File transfer time in ms (negative if error)

Status File The software must record in real time, in a single file, the progress of the backup jobs. The minimum expected information is:

- Timestamp
- Naming the backup job
- Backup job status (e.g. Active, Not Active...) If the job is active
 - o The total number of eligible files
 - o The size of the files to be transferred
 - o The progression
 - Number of remaining files
 - o Size of remaining files
 - Full address of the Source file being backed up
 - o Complete address of the destination file

The locations of the two files described above (daily log and status) will have to be studied to work on the clients' servers. As a result, "c:\temp" type locations are to be avoided.

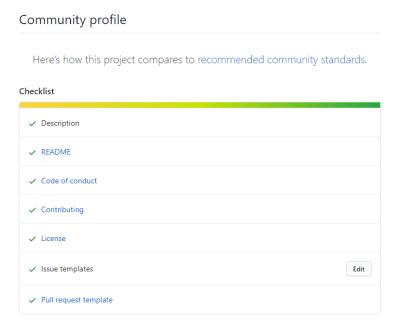
The files (daily log and status) and any configuration files will be in XML or JSON format. In order to allow fast reading via Notepad, it is necessary to put line feeds between the XML (or JSON) elements. A pagination would be a plus.

Project Management

In order to allow each member of our team to create and participate in the most efficient way possible we had to put together a project management strategy. This strategy translates into our collaborative work via the GitHub platform, where our code and the use of design templates such as the VCM have allowed us to easily distribute tasks. Moreover, in a concern of perenniality of the application, these design patterns will allow us to pass in a second time to a graphic interface in a much simpler way.

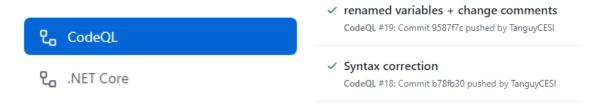
Beyond GitHub we also collaborated via Azure Devops via a board, adding To Do list so that everyone knows the remaining tasks and have a simpler and clearer vision of our Github repository. But also to set up pipelines to certify that our code is working properly and that we are using good practices during this project.

To create a project we definitely wanted to respect every standard given by our tools. In GitHub we respected each and every recommandation as you can see in the following screen:



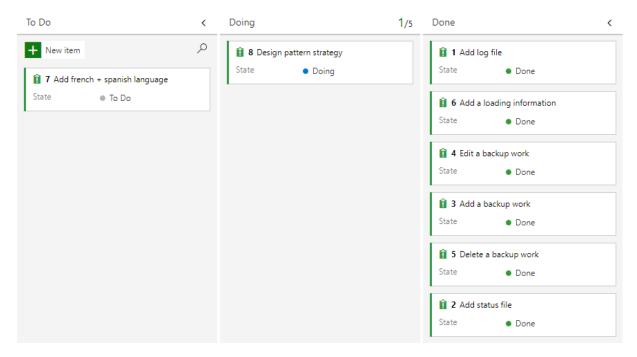
Each file can be found in annex at the end of this document.

We did not have to neglect security. Thus our only dependency is <u>Newtonsoft.Json</u>, but to go beyond and add more security to our project we have gone further by adding a pipeline which on each single pull / push will check the code quality (based on given C# rules) and report any anomaly.

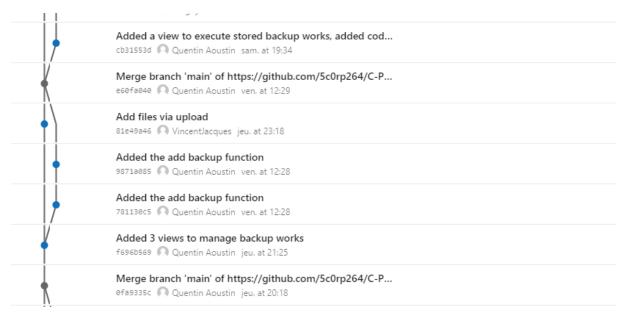


Example of Code Quality check based on our commits.

Finally we also used Azure DevOps to manage our project. We mainly used the board functionality to list our tasks.



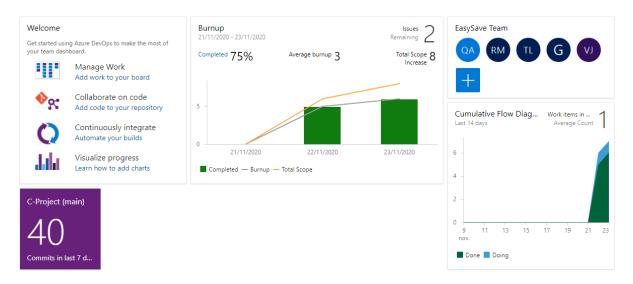
We also linked our repository to set up and manage it easily from the azure devops dashboard. Furthermore it gave us the possibility to follow visually the commits made by each member with a timeline element.



Finally, this dashboard gave us the possibility to track manage and add branches.



We add full power over our repository with team management, a real plus was the ability to track our speed in order to respect all of our deadline with charts tool present in the dashboard.



To conclude, Azur DevOps, GitHub, and Visual Studio 2019 (which include a GitHub extension) were all tools to manage this project. We really saw a benefit here task distribution was really simplified thanks to those tools.

To be concrete here are the task made by each member of the EasySave project:

We will refer as "main task" the task for which the team member has spent the most time. In fact, everyone was able to make progress on each task.

Team member	Main task
Tanguy Lhinares	Code + documentation
Aoustin Quentin	Code + documentation
Jacques vincent	UML
Jérémy Gabriel	UML
Rémi Mounier	UML

We did not use GitHub desktop during this project only Git Bash to clone, pull, add, commit and push our project / modifications.

The management of this project was pushed even further by the appliance of Agile Method, as we could see in the Azure DevOps dashboard with burnup charts. The reusage of our code, usage of Design Patterns and timeboxing in fact one of our priority was to respect the deadlines imposed for the first version to honor this we used timeboxing, this way everyone could respect the due date of the diagrams and the code.

EasySave and Documentation

EasySave is not just a project, it is the future of every backup software you will not need any other one after using ours.

Accompanied by meticulous documentation and multiple guides to collaborate with us in this project, report an issue and clear rules about contributing and finally with a security file to keep you updated with security you will find every answer to your questions.

We will now present each feature detailed in the <u>Introduction</u> of this document. But in order to follow this part you may need to respect some prerequisites to make this project functional.

First of all you need to install Visual Studio with .NET Core 3.0 in order to run the program and check our code.

Once everything is installed clone the repository in the folder of your choice via the following command:

git clone https://github.com/5c0rp264/C-Project

You may need to delete the db.json file at the root of the project. Its position is an arbitrary choice of facility for the user as for the state file and the log one detailed hereafter. This position is static and prevent the user to be lost between is temporary memore in the %appdata% folder or wherever. Every single file will be stored in the root of our project.

If the software is used in a company it adds an easier way of management in case of necessary administration the path will always be the same.

Now you can open the .exe file of the first release version. A tailor-made polished console will then open.



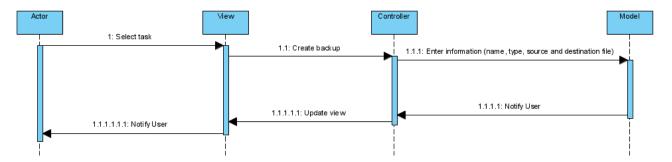
You will find:

- 1. Our logo and the title of the application
- 2. A close button + a scroll bar in case of need
- 3. Our console designed logo
- 4. A menu to make your choice based on identifiers (0 / 1 / 2 / 3)
- 5. An input to enter the choice you want to make.

We will now perform in-depth looking of each feature.

Adding a backup job

How does adding a backup job works in our code?



Sequence diagram: Adding a backup job

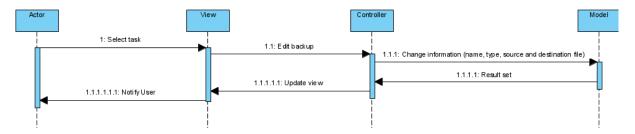
When we translate this in our console, the software will ask succinctly for each characteristic.



Console view of EasySave when we add a new backup job

Editing a backup job

How does editing a backup job works in our code?



Sequence diagram: Editing a backup job

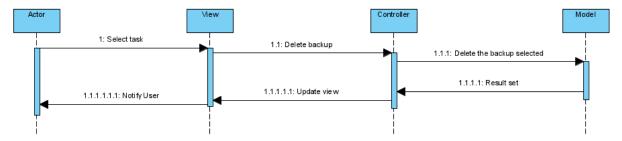
When we translate this in our console, the software asks for each criteria of the backup job and can modify it (here we change the name and the type (from full to differential)) and we just press enter to automatically refill the previous criteria with the previous value. A memento design pattern is here implemented with the externalisation of the data to conserve them even when change happens, this way we can always return to a previous state.



Console view of EasySave when we edit a backup job

Deleting a backup job

How does deleting a backup job works in our code?



Sequence diagram: Deleting a backup job

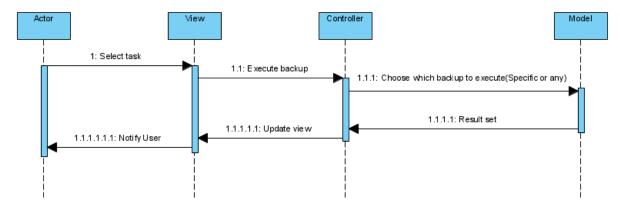
When we translate this in our console, the software shows a table linking each backup job with an ID, the display is the same as for the editing feature. Here we will choose the only available backup job.



Console view of EasySave when we delete a backup job

Deleting one/multiple backup job(s)

How does executing one or multiple backup job(s) work(s) in our code?



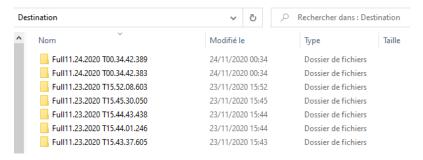
Sequence diagram: Executing (a) backup job(s)

When we translate this in our console, the software shows a table linking each backup job with an ID, the display is the same as for the editing and the deleting features. Here we will choose to execute the first and the second backup jobs

```
EasySave from ProSoft
                                                                                                                              П
                                                                                                                                     ×
[0] Execute backup jobs
[1] Add a backup job
[2] Edit a backup job
[3] Delete a backup job
Your choice :
[Id]
[1]
[2]
        Backup1
        Backup2
[4]
[5]
        Backup4
        Backup5
Id of backup job you want to execute :
Do you want to add other backup job [0]No 1[Yes] :
Id of backup job you want to execute :
Do you want to add other backup job [0]No 1[Yes] :
Press a key to continue
```

Console view of EasySave when we delete a backup job

In our destination folder we have each full backup executed before. The 2 previous one are those with the datetime 24/11/2020:



Preventing user's mistakes

Each feature demonstrated among this part is correctly made with check. In fact one of the criteria is to limit the user to 5 backup job. We could image that the user does not remember his number of backup or try to bypass this limit.



He can also try to execute a backup which does not work in this case each exception throwed will be shown:

```
EasySave from ProSoft
                                                                                                                    Х
[0] Execute backup jobs
[1] Add a backup job
  | Edit a backup job
[3] Delete a backup job
Your choice :
[Id]
        Name
        Backup1
       Backup2
       Backup3
[4]
[5]
       Backup5
Id of backup job you want to execute :
Do you want to add other backup job [0]No 1[Yes] :
Full backup of reference for diff backup [3] Backup4 :
-Source or full backup directory does not exist or could not be found: L <
```

Here as we can see, the destination does not exist, and the full backup of reference does not exist too.

Finally if the user does not have any backup job he can not edit / delete one.

```
Your choice :
3 Your choice :
2
You don't have any backup job to delete...
Press a key to continue
Your choice :
2
You don't have any backup job to edit...
Press a key to continue
```

In case of invalid input the view is refreshed to add a warning message:

```
What type of backup do you want :
[0] Differential
[1]Full
3
Thanks to enter a valid value.
```

Log and State files

As required in the subject of the project we created two files.

The state file which is an exhaustive file detailing the content of a backup with the size of each file, their path, size of remaining files, progress, backup job details (name, source, destination, type), if the job is still active or not, and even more information that we were not asked for, but we decided to add despite all.

You can see an example of this file when we launch a backup job with a source folder containing a zip file, a file with nothing in it and a "picture" folder with two pictures.

```
state.log - Bloc-notes
Fichier Edition Format Affichage Aide
    "Name": "MyFiles",
"Source": "C:\\Users\\Tanguy Lhinares\\Desktop\\Source",
"Destination": "C:\\Users\\Tanguy Lhinares\\Desktop\\Destination",
"FolderName": "/Full1.24.2020 T01.01.58.635",
      TotalElligibleFile": 4,
     "ElligibleFiles": [
       "fileSize": 142879,
         "fullName": "C:\\Users\\Tanguy Lhinares\\Desktop\\Source\\abecedary.zip"
      "fileSize": 0,
"fullName": "C:\\Users\\Tanguy Lhinares\\Desktop\\Source\\test.txt"
         "fileSize": 178296,
"fullName": "C:\\Users\\Tanguy Lhinares\\Desktop\\Source\\picture\\work1.jpg"
         "fileSize": 121117,
"fullName": "C:\\Users\\Tanguy Lhinares\\Desktop\\Source\\picture\\work2.jpg"
    ],
"IsFull": true,
"ISACtive": false,
     "FilesTransfered": [
         "fileSize": 142879,
"fullName": "C:\\Users\\Tanguy Lhinares\\Desktop\\Source\\abecedary.zip"
         "fileSize": 178296, "fullName": "C:\\Users\\Tanguy Lhinares\\Desktop\\Source\\picture\\work1.jpg"
         ],
"Stopwatch": {
  "IsRunning": false,
  "Elapsed": "00:00:00.0777763",
  "ElapsedMilliseconds": 77,
  "alansedTicks": 777763
      'Progress": 1.0
     "TotalSizeOfElligbleFiles": 442292,
     "SizeOfRemainingFiles": 0
```

Please note that the destination contains the exact same files than the source. The variable ISActive is set to false because the backup job took less than one second to finish.

Finally as demanded the file extension is parsed in json, but it was not really readable, so we added automatic indentation.

The log file which is an exhaustive file detailing each backup job executed the same day because this log file has to be a daily one. Today we launched only one backup job, so the log file is not really exhaustive.

```
[
   "timestamp": 1606176118.6611748, Timestamp
   "message": "Starting the backup job [1] MyFiles"
                        Naming the backup job
   "timestamp": 1606176118.6696537,
   "message": "Source : \\?\\C$\\Users\\Tanguy Lhinares\\Desktop\\Source"
      Source / Destination full address + UNC format
   "timestamp": 1606176118.6738124,
   "message": "Destination : \\?\\C$\\Users\\Tanguy Lhinares\\Desktop\\Destination"
   "timestamp": 1606176118.678552,
   "message": "Source directory size : 442292"
   "timestamp": 1606176118.682489,
   "message": "Destination directory size : 442292"
           Directory size + transfer time
   "timestamp": 1606176118.7387507,
   "message": "Transfer time : 00:00:00.0777763"
 }
]
```

Please note that we did not wrote the file size but the directory size, as we saw before each file is already detailed in the state file.

The last thing required was to create JSON elements, like for the state file, once again we added automatic indentation.

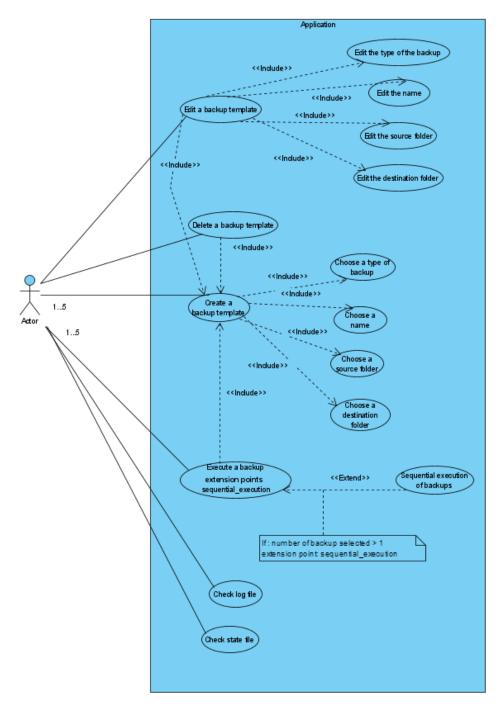
Unified Modeling Language - Diagrams

One of our deliverables is composed of multiple UML diagrams. In the previous sections, you saw our sequence diagram which explains the different actions and retrace the decision made by our program. In addition to those four diagrams we have four others.

Nota bene: if the quality of the diagrams is not sufficient please follow this link.

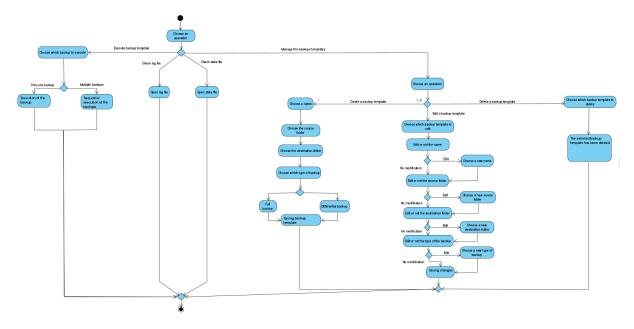
The use case diagram

A perfect fit to the documentation is the use case diagram which aims to explain the relation between the actor (you or any other user) and the application.



The activity diagram

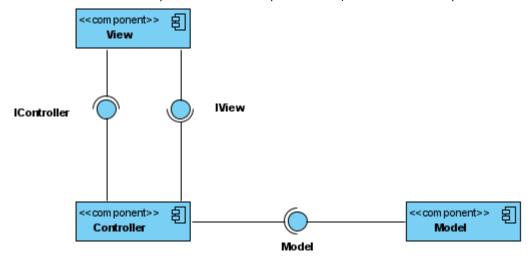
The activity diagram is a UML behavioral diagram, allowing to represent the triggering of events according to system states and to model parallelizable behaviors.



The component diagram

Before introducing the class diagram we should look over the architecture of EasySave.

Based under the Model View Controller (MVC) design pattern we have 3 essentials component, this diagram shows the controller as the link between the view (what is displayed to the user) and the model (where we process the data).

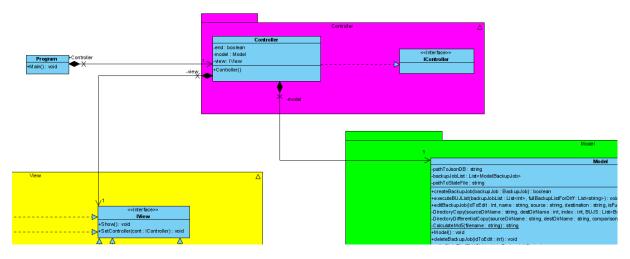


The class diagram

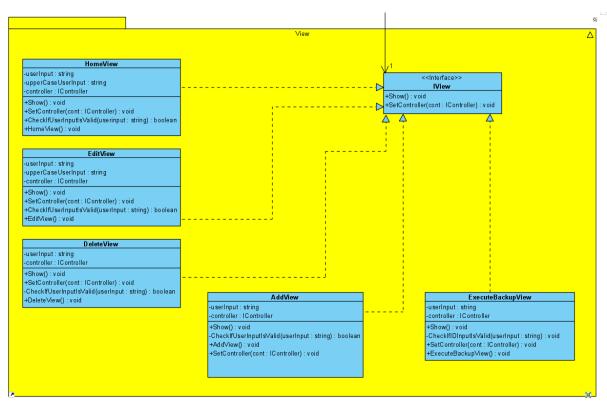
Finally the class diagram is a really exhaustive one where you will see each class, with their respectives attributes and method, relations and multiplicity is also apparent in our diagram. Each class fits into a folder between the 3 folders we have respectively for the model, the view, and the controller but we will not start in this order.

This diagram being really huge we will show each folder one by one starting in the order of the program:

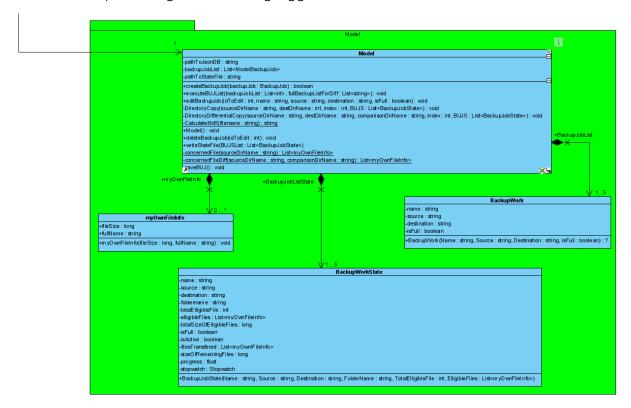
First of all a controller is created by the Program.cs file which is not present in any folder, it is the root file of our software.



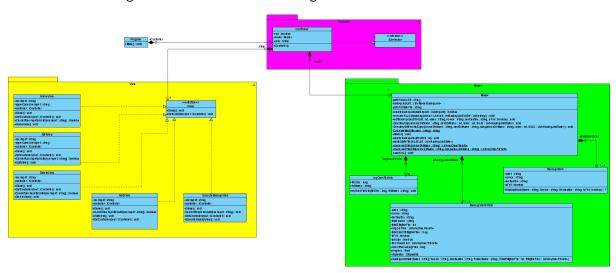
The controller instantiate a new view to display the main menu:



Finally our view receives input transmitted to the controller, those inputs are then transferred to the model for processing data, creating logg and state files...



All those folders together leads to this main diagram:



Visual Paradigm files for each diagram:

Sequence diagrams => GitHub Link

Use Case diagram => GitHub Link

Activity diagram => GitHub Link

Component diagram => GitHub Link

Class diagram => GitHub Link

Conclusion

After having created UML diagrams specific to what we wanted for our code, our vision was much clearer, the implementation of a global management system (Azure DevOps, Visual Studio 2019, Git Bash) allowed us to distribute the tasks and meet all the expectations of this first deliverable. The lockdown set up due to the sanitary situation did not prove to be a difficulty thanks to the good use of the appropriate tools presented above and thanks to the rapidity of answers from our teachers, whom we thank again.

Our team has invested a considerable amount of time in this first release and therefore hopes that it will bring satisfaction to MiProSoft for future improvements. As the requested specifications have been fully respected, our skills in project management, UML diagramming and development have been honed through this deliverable.

Personal project review:

LHINARES Tanguy

> This project was carried out thanks to the cooperation of all the members of the group and their availability even outside the hours normally dedicated to this deliverable, the discovery of some tools like Azure DevOps or pipelines in GitHub was a pleasure and a real added value for my skills, I really enjoyed making these diagrams and especially this console version of EasySave. I hope the next release will bring as much.

AOUSTIN Quentin

JACQUES Vincent

GABRIEL Jérémy

MOUNIER Rémi

Apendixes

Annex 1: CODE_OF_CONDUCT.MD

Contributor Covenant Code of Conduct

Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- · Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- · Focusing on what is best for the community
- · Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- · The use of sexualized language or imagery and unwelcome sexual attention or advances
- · Trolling, insulting/derogatory comments, and personal or political attacks
- · Public or private harassment
- · Publishing others' private information, such as a physical or electronic address, without explicit permission
- · Other conduct which could reasonably be considered inappropriate in a professional setting

Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at tanguy.lhinares@viacesi.fr. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

Annex 2: CONTRIBUTING.MD

Contributing to EasySave

We love your input! We want to make contributing to this project as easy and transparent as possible, whether it's:

- Reporting a bug
- · Discussing the current state of the code
- Submitting a fix
- Proposing new features

We Develop with Github & Azure Devops

We use github to host code, to track issues and feature requests, as well as accept pull requests. If you want to be added to our Azure project, feel free to contact us.

We Use Github Flow, So All Code Changes Happen Through Pull Requests

Pull requests are the best way to propose changes to the codebase (we use Github Flow). We actively welcome your pull requests:

- 1. Fork the repo and create your branch from master.
- 2. If you've added code that should be tested, add tests.
- 3. If you've changed APIs, update the documentation.
- 4. Ensure the test suite passes.
- 5. Make sure your code lints.
- 6. Issue that pull request!

Any contributions you make will be under the MIT Software License

In short, when you submit code changes, your submissions are understood to be under the same MIT License that covers the project. Feel free to contact the maintainers if that's a concern.

Report bugs using Github's issues

We use GitHub issues to track public bugs. Report a bug by opening a new issue; it's that easy!

Great Bug Reports tend to have:

- · A quick summary and/or background
- Steps to reproduce
 - Be specific!
 - Give sample code if you can.
 - Includes sample code that anyone at least in our team can reproduce with the minimal setup what
- What you expected would happen
- What actually happens
- Notes (possibly including why you think this might be happening, or stuff you tried that didn't work)

People love thorough bug reports. I'm not even kidding.

Use a Consistent Coding Style

I'm again borrowing these from Facebook's Guidelines

- 2 spaces for indentation rather than tabs
- Any code you provide should respect camel case syntax and good C# practice

License

By contributing, you agree that your contributions will be licensed under its MIT License.

References

This document was adapted from the open-source contribution guidelines for Facebook's Draft

Annex 3: LICENSE

```
MIT License
3
   Copyright (c) 2020 5c0rp264
5
   Permission is hereby granted, free of charge, to any person obtaining a copy
   of this software and associated documentation files (the "Software"), to deal
    in the Software without restriction, including without limitation the rights
   to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
    copies of the Software, and to permit persons to whom the Software is
    furnished to do so, subject to the following conditions:
12
    The above copyright notice and this permission notice shall be included in all
    copies or substantial portions of the Software.
13
14
   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
15
    IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
   FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
   AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
   LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
   OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
    SOFTWARE.
```

Annex 4: PULL_REQUEST_TEMPLATE.MD

Please ensure your pull request adheres to the following guidelines:

Use the following format: * owner/repo Link additions should be added to the bottom of the relevant category. New categories or improvements to the existing categorization are welcome. Search previous suggestions before making a new one, as yours may be a duplicate. Sort by alphabetical order Thanks for contributing to EasySave!

Annex 5: SECURITY.MD

Security Policy Supported Versions Our currently secured versions of EasySave Version Supported 1.0 2.0 Not released yet 3.0 Not released yet

Reporting a Vulnerability

If you have any issue please go to the corresponding section (see in the readme.md file) and respect the issue template in our .github folder. Thanks for contributing to EasySave I

Annex 6: bug_report.md

name	about	title	labels	assignees
Bug report	Create a report to help us improve			

Describe the bug A clear and concise description of what the bug is.

To Reproduce Steps to reproduce the behavior:

- 1. Go to '...'
- 2. Click on '....'
- 3. Scroll down to '....'
- 4. See error

Expected behavior A clear and concise description of what you expected to happen.

Screenshots If applicable, add screenshots to help explain your problem.

Desktop (please complete the following information):

- OS: [e.g. iOS]
- · Browser [e.g. chrome, safari]
- Version [e.g. 22]

Smartphone (please complete the following information):

- Device: [e.g. iPhone6]
- OS: [e.g. iOS8.1]
- · Browser [e.g. stock browser, safari]
- Version [e.g. 22]

Additional context Add any other context about the problem here.

Annex 7 : feature_request.md

name	about	title	labels	assignees
Feature request	Suggest an idea for this project			

Is your feature request related to a problem? Please describe. A clear and concise description of what the problem is. Ex. I'm always frustrated when I

Describe the solution you'd like A clear and concise description of what you want to happen.

Describe alternatives you've considered A clear and concise description of any alternative solutions or features you've considered.

Additional context Add any other context or screenshots about the feature request here.

Annex 8: Worflows -> codeql-analysis.yml

```
name: "CodeOL"
on:
 push:
   branches: [ main ]
 pull_request:
   # The branches below must be a subset of the branches above
   branches: [ main ]
  schedule:
   - cron: '40 18 * * 3'
jobs:
  analyze:
   name: Analyze
   runs-on: ubuntu-latest
   strategy:
     fail-fast: false
     matrix:
       language: [ 'csharp' ]
       # CodeQL supports [ 'cpp', 'csharp', 'go', 'java', 'javascript', 'python' ]
       # Learn more:
       # https://docs.github.com/en/free-pro-team@latest/github/finding-security-vulnerabilitie
    steps:
    - name: Checkout repository
     uses: actions/checkout@v2
    # Initializes the CodeQL tools for scanning.
    - name: Initialize CodeOL
     uses: github/codeql-action/init@v1
     with:
       languages: ${{ matrix.language }}
       # If you wish to specify custom queries, you can do so here or in a config file.
        # By default, queries listed here will override any specified in a config file.
        # Prefix the list here with "+" to use these queries and those in the config file.
        # queries: ./path/to/local/query, your-org/your-repo/queries@main
    # Autobuild attempts to build any compiled languages (C/C++, C#, or Java).
    # If this step fails, then you should remove it and run the build manually (see below)
    - name: Autobuild
     uses: github/codeql-action/autobuild@v1
    # i Command-line programs to run using the OS shell.
    # 😝 https://git.io/JvXDl
    # / If the Autobuild fails above, remove it and uncomment the following three lines
       and modify them (or add more) to build your code if your project
       uses a compiled language
    #- run: |
    # make bootstrap
    # make release
    - name: Perform CodeQL Analysis
     uses: github/codeql-action/analyze@v1
```