

PARTIFY

Group project report

Group 3:

NOGARET Baptiste
BOJARSKI Marcin
SMOLARCZYK Maciej
RAZAFINDRAMANANA Anthony

Table of content:

[Requirement](#)

[Presentation of the group](#)

[Tools](#)

[Meetings](#)

[Personal work](#)

[Post project review](#)

[How did we split the work and manage our tasks ?](#)

[What went well during this group project and bad and if we have to do it again what would we change ?](#)

[What is the result ?](#)

[How far did you go from our expectations ?](#)

[What we learnt - Maciek](#)

[What we have learnt](#)

[Idea of improvements - Baptiste](#)

[Appendices](#)

[State of the art](#)

[Screenshot of github page with our application logo](#)

[How we progress working on the project\(commits to days\)](#)

[Mockup of our application](#)

[Screenshot of the kanban page on taiga.io](#)

[Screenshot of the wiki page on taiga.io](#)

TODO: Introduction what the project was about.... Our goals - **Anthony**.

Requirement

TODO: explain what it is about - **Baptiste**.

MUST:

- Fancy Gui (ergonomic, modern look, intuitive, drag and drop?)
- Desktop (cross platform)
- Metis (+ optional parameters for advanced)
- Our intermediate format (convert)
- Visualise the result (graph / mesh) with different colors / shapes for each partition (2D - 3D?)
- Performance measurements : edge-cuts, load balancing, time
- Recommendations (method, ...) :
 - method description with the advanced mode
- Benchmark : compare solutions : array with edge-cut, load balancing, computational time

SHOULD:

- Parmetis
- Remote server :

- Connect to astral
- Upload / download file to astral (compress / uncompress the file without loss)
- Execute metis / parmetis on astral
- Zoom and move the visualisation
- Save the visualisation
- Add format for input (Shaco <-> Intermediate) {Easiest implementation}
- Use other libraries (shaco, ...)
- 3D

CAN:

- Use the queue of astral
- “Management” tool for the queue (queueing work, when it will be executed, delete, add, number of process ...)

Presentation of the group

- Three of us had already work together for requirement analysis on scientific paper presentation.
- Half polish, half french train working in international team
- We all live on the campus.
- Only smart hard workers are chosen into our team.
- Polish knew each other from the same university.
- A group of nice guys, focus on the success of the project.
- Different educational backgrounds

Tools

We are using git for versioning control. It is hosted on github and can be found [here](#).

We were using Taiga that is a project management platform for agile developers. It implements the “kanban” methodology.

It had also a “wiki” section there. We used it to share tutorials to Javascript or Git, compiled libraries and or libraries.

As it was for most of the members the first project using Javascript, we read to follow the language’s convention. For example, we read with dedications the [Google JavaScript Style Guide](#) that enounces some common language and style rules.

What’s more, we search on github some projects using Node.Js (and especially Electron such as [VsCode](#) or [Atom](#) project) to copy the way they structure their applications. That’s why we used a linter to avoid some common mistakes and have the same coding norms. All the existing rules are available on the [eslint website](#). The list of precepts that we used for our projects can be found in our github directory ([link](#)).

Meetings

Meeting date	25.01.2016
Time	19:45 - 20:45
Participants	All team
Agenda	<ol style="list-style-type: none"> 1. Discussion about possible tools, programming language we want to use 2. Discussion about libraries we may use. 3. Sharing tasks 4. Organizing time and set time of next deadline and current tasks.
Notes	<p>Our meeting was mostly about researching available tools, libraries and programming languages. We share responsibilities for each part of research and created appropriate tasks on Kanban board.</p> <p>We created tasks:</p> <ol style="list-style-type: none"> 1. Investigate possible tools(C#, C++, JS) 2. Check how stand-alone METIS version works 3. Check other libraries input and output and compare them. 4. Check for some academic research papers about graph partition.

Meeting date	02.02.2016
Time	11:50 - 12:50
Participants	All team
Agenda	<ol style="list-style-type: none"> 1. Gather a results of tasks we set at previous meeting. 2. Choose the technologies (tool, language, library ...)
Notes	<ol style="list-style-type: none"> 1. See the annexe for the sum-up of pros and cons of the technologies. Finally, we decided to use Electron. 2. We chose Electron + JS and SigmaJS. 3. We can try to design UI using Balsamiq tool in order to have an overview of the functionalities of our final solution. 4. Launch graph partitioning library on Windows. 5. Check for unit test framework. 6. Check if we should follow some standard coding

	7. Check for idea of GUI
--	--------------------------

Meeting date	03.02.2016
Time	10:40 - 12:00
Participants	All team
Agenda	1. Did we manage to run any library on Windows and make decision which one to use.
Notes	<ol style="list-style-type: none"> 1. We were not able to build and run Kahip and Chaco on Windows. 2. Marcin and Maciej were able to build and run MeTiS example on Windows. 3. We decided to start developing both on windows and linux using MeTiS library. 4. We decided to use the stand-alone metis. Code to call it from electron. 5. Using sigma and d3.js, display the graph that metis use as an input file (need a conversion phase) 6. Try to use parmetis from windows.

Meeting date	04.02.2016
Time	20:15 - 21:00
Participants	All team
Agenda	Discuss progress and results from previous meeting. Working on occurred problems i.e. running NodeJS on Maciej's PC. Help Marcin to run parMetis.
Notes	We decided that all of us will continue the job, apart from that Baptiste will try to connect to remote server(Astral).

Meeting date	07.01.2016
Time	15:45 - 17:00
Participants	Anthony, Baptiste, Maciej
Agenda	We discuss about integrating stand-alone library results with our plotting library.

Notes	<p>We compared our solutions of rendering graphs. Together we managed to change color of particular nodes. We decided to check if there is a possibility to improve the performance of VivaGraph.</p> <p>Marcin participation was not necessary today.</p>
-------	--

Meeting date	08.02.2016
Time	11:30 - 12:15
Participants	All the team
Agenda	<p>Discuss progress on:</p> <ul style="list-style-type: none"> - integration of parmetis - displaying the graph and mesh and coloring them according to the result files of metis - connection to a remote server
Notes	<p>Anthony will prepare a ready solution to draw graph and mesh.</p> <p>Marcin and Maciej will work on parmetis issue.</p> <p>Baptiste will continue with the issues of connecting to the server.</p>

Meeting date	09.02.2016
Time	20:15 - 22:30
Participants	All the team
Agenda	<p>Discuss the user interface:</p> <ul style="list-style-type: none"> - discuss the gui mockup - check different css framework <p>Running parmetis with exe file and its different parameters (8 parameters)</p>
Notes	<p>Decide to choose Bootstrap framework</p> <p>Baptiste, Anthony and Maciej are going to work on the gui and merging our parts (graph, remote connection and executing stand-alone)</p> <p>Marcin is going to investigate in details how to use parmetis (input, metric to measure the quality, parameters, output)</p>

Meeting date	12.01.2016
Time	11:00 - 12:45
Participants	All team
Agenda	<p>Different outputs from Metis and ParMetis - what to do with that?</p> <p>Running metis and parmetis executable files on different machines with Windows through our GUI.</p> <p>Discuss communication problems between front-end and back-end.</p>
Notes	<p>Baptiste, Anthony and Maciej are going to work on:</p> <ul style="list-style-type: none"> • GUI • remote server • integrate rendering part with UI • put window for choosing library and some options for execution file <p>Marcin will work on:</p> <ul style="list-style-type: none"> • add time measuring in parMetis • set different value of maximum imbalance • gather the results from parMetis.exe and metis.exe and create some plots/table using different graphs/meshes.

Meeting date	16.02
Time	18:00 - 18:35
Participants	All team
Agenda	What type of input parameters we would like to put to GUI and the execution of each library.
Notes	<p><u>parMatis:</u> Input: number of processors, graph file, number of parts, maximum unbalance.</p> <p>mandatory fields: number of processors, number of parts Output: time, balance, number of cuts.</p> <p><u>Metis Graph:</u> Input: graph file, number of parts, maximum unbalance, partitioning type, coarsening type, initial partitioning type, objective type, u-factor(change to ubvec!), number of iteration(niter).</p>

	<p>mandatory fields: number of parts</p> <p>Output: load imbalance, number of cuts, time, memory.</p> <p><u>Metis Mesh:</u></p> <p>Input: graph file, number of parts, maximum unbalance, partitioning type, coarsening type, initial partitioning type, objective type, u-factor(change to ubvec!), number of iteration(niter).</p> <p>Output: load imbalance, number of cuts, time, memory.</p> <p>*kway - objectype, *recursive bisection require only: iptype,</p> <p>Marcin is going to change the output of both libraries to make easier to parse their output. Maciej, Baptiste and Anthony will continue with GUI. Maciej: get parameters from user and execute file with them. Anthony: display library output. Prepare main page with example graph and application name. Optionally: Add "pause" button for graph animation. FPS measurements. Baptiste: working on remote server.</p>
--	---

Meeting date	04/03/2016
Time	02.15 - 3.20 pm
Participants	Baptiste - Anthony - Maciej
Agenda	Report progress
Notes	<p>Features:</p> <ul style="list-style-type: none"> - Chaco must be tested + display the performance - Changes the size of the nodes for mesh. - Show performance when we use remote servers. - Bug parmetis <p>GUI:</p> <ul style="list-style-type: none"> - center the results in footer(change them to more fancy) (B) - display links and colors buttons -> set absolute position (M) - add hovers to Start/Pause buttons (M) - button for hide/show footer. (B) - add logo (A) - change fonts in app name, and in about section.(A) - change color of application. (?) - add dialog backgrounds (B) - change examples to more 'fancy' (A)

	<ul style="list-style-type: none"> - create content for the backgrounds (M) <p>Features for the presentation</p>

Meeting date	07.03
Time	20.15 - 22.15
Participants	All team
Agenda	<p>Preparation of the presentation: skeleton</p> <p>TODO for videos:</p> <ul style="list-style-type: none"> - Colors (B) - Show the number of edges and vertices (A) - Switch when we display the colors (A) - "Help manual" for the option dialog windows (MAC) - CHACO (B)
Notes Can use	<p>Introduction:</p> <ul style="list-style-type: none"> • Who can use it (who is-it for) <ul style="list-style-type: none"> ◦ Non-technical users ◦ Advanced users • Introduce our company / introduce ourself • What problem can the app solved • Introduce our product <p>Presentation in more details of our apps:</p> <p>Advantages of our apps (compare to the others)</p> <ul style="list-style-type: none"> • Buy server on the cloud: our own server, astral • It is cross platform (linux and windows) • it works with 3(!) libraries: chaco, metis, parmetis: fastest library • Flagship product • User may run calculations in parallel to get the results faster • Advantages of our app compared to the competitor • Desktop application can be used without internet. • Use remote server in case the desktop isn't enough powerful (or use more processors). • Can execute library on a remote server • Visualization of the graph / mesh: zoom in / out, move the graph (2D) • User-friendly

	<ul style="list-style-type: none"> • Executable available • Can execute it on any server • Information about the partitions: edge cuts ... <p>Feature to present on video/gif:</p> <ul style="list-style-type: none"> - Zoom in and out + move the graph - show graph without and with colours + show graph without and with edges - Fill out the forms (switch metis / parmetis -- type wrong values -- <p>Features to present on slide:</p> <ul style="list-style-type: none"> - calculations result on the black bar
--	--

Meeting date	08.03
Time	20.15 - 23.15
Participants	All team
Agenda	<p>Prepare presentation:</p> <ul style="list-style-type: none"> - prepare the slides - Know what to present in the videos <p>Corrections of some bugs</p>
Notes	

Meeting date	09.03
Time	10.30 - 13.30
Participants	All team
Agenda	<p>Finishing the slides</p> <p>Training a few times</p>
Notes	

Personal work

	Meeting work (12 meetings)	Personal work
NOGARET Baptiste	20h45	Around 85h
RAZAFINDRAMANANA Anthony	20h45	Around 80h
BOJARSKI Marcin	18h25	77h03m57s
SMOLARCZYK Maciej	20h45	80h

Post project review

How did we split the work and manage our tasks ?

What went well during this group project and bad and if we have to do it again what would we change ?

First of all, according to our research and our “must” goals (before starting the project) we were agree to go through web technologies (javascript, html5, css3), although only Baptiste already used it. Also since three of us already have windows installed on our computers, we developed mainly our software using Microsoft’s Metis and parMetis libraries nonetheless Baptiste has a linux computer and manage to in parallel add functionalities to the software and use it on linux. Marcin by being the only one able to properly install Metis and parMetis library on Windows becomes our graph partitioning library expert. He tested and played with the library meanwhile Maciej, Baptiste and Anthony build the application based on Electron. Even if all of us work over all applications they specialized on main features. Maciek worked mainly on the options dialog in order to allow the user to enter the differents parameters and execute library using different options. Baptiste took care of the connection to the remote servers, install extend library and monitor the good behavior on Linux. Whereas Anthony deals with the visualisation part and the main windows. But as underlines we need to communicate as they are a lot of dependencies within the software itself and through the library. Also we used as a source code management SourceTree, this way we efficiently divided our works and then during meeting (and unreported meeting) merged everything and advanced step by step during all project.

However at the beginning we had to take back some modifications as it had conflict with some functions implement by others members and also because we weren’t all familiar with source code managers. But after knowing better each others and the fundamentals of the software we managed when task are given to update separatly the softwares. One of our mistakes could be to have not defined precisely how every single implementation worked so when we add features instead of reviewing code already implemented we based on it to produce new features and at the end we have stack of working code which are difficult to add additional features.

ParMetis library is one a good example, because we can run the library but we didn’t realise that name of the output are different and to set it we have to tack back our code from the beginning, and one it could avoided by send the stand alone that we executed instead of using all the trick already implemented within our software.

What is the result ?

However our software is quite complete, we manage to implement three libraries (including Chaco on linux) and get output of them including key performances indicators and the partitioning in order to display it to the users. In addition the rendering is nice, we can reach even more than 16 000 points (depends of course of the computers). Also we can access to remote server. And finally we provide a nice user interface with lots of details features that you may not really notice but makes our application user-friendly (save image, example of graph and mesh, access to manual, notifications).

How far did you go from our expectations ?

If we now compare with the “must”, “should” and “can”, that we have forecast before the beginning of the group project we reach almost all of the expectations. The following table depict how far did we go and what remained to be implemented: (some of them have been renamed according to the current context)

Level of expectations	Expectations	Reach %
Must	Fancy Gui	96% (Could be fancier)
	Desktop version	100%
	Metis	100%
	Output reader	80% (Can't read parmetis output)
	Visualise the result	95% (Show biggest graph or mesh)
	Library performance	100%
	Recommendation	90% (Maybe more personalized to the user rather than provide the user manual)
	Benchmark	0%
Should	Parmetis	75% (only able to execute it not to display it but we already know the problem but we have to make from the beginning our apps)
	Remote server	100% (try only with astral)
	Zoom and move the visualisation	100%

	Save the visualisation	100%
	Reader file for shaco	
	3D	96% (using 4elt.graph and tiny mesh or graph we can get 3D results but the only big mesh file that we can display is metis.mesh and it doesn't look good).
	Add other libraries	70% (Chaco and not fully tested)
Can	Using astral queues	0%
	Management tool for the queue	0%

TODO: Library performance, short description provided by [Marcin](#).

What we have learnt

During the project we encountered many challenges. They were both technical and interpersonal. We were working in many different roles like software engineers, testers, marketing, researchers, designers, UX or management. According to that we had many opportunities to familiarize with new tools for project management, code versioning, creating mockups or learn fundamentals and some more advanced aspects of web development. Working on the project we could not forget that we have to work as a team and all of us are responsible for success of the project in the same way. It is necessary to have a plan and some assumptions before start working. We should be sure that all of team members know what we are going to do because misunderstanding of one person may complicate further work of other members and delay project progress.

We had many opportunities to discuss with each other during whole time of the project. It was important to be able to present our ideas, got critics from someone and give our opinion about other team members concepts. After every discussion, we had to take a joint decision what also require abilities to convince someone to our idea.

Finally we did some marketing part of our project like create a name, logo and presentation of our application.

Thanks to this project each of us had opportunities to see how it is to work in many different roles and verify which of them we like to do. Additionally we know which our individual skills were weak and should be improved in the near future.

Idea of improvements - Baptiste

FOR MARCIN:

- install NodeJS
 - Instal GIT
 - Clone project from git
 - Npm install
 - Install mpi(mpich)
 - Instal metis, parmetis, chaco - instal default versions

Appendices

State of the art

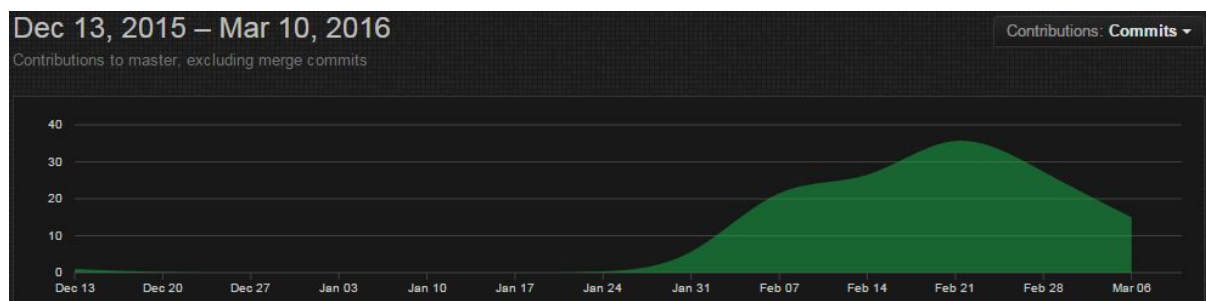
Since we set our general goals for the project, we did a research among tools which may help us to build the application in efficient way. Main factors which we considered were possibility of drawing graphs, creating fancy user interface and our programming background. To make better choice we gathered all pros and cons in a table below.

Framework	Optional library	Pros	Cons
Unity 3D		Good community support. A lot of tutorials and examples. Dedicated for rendering graphic 3D. Development in C#.	Doesn't support development on linux. Need to learn the tool from very beginning. More video games oriented. None of us has experience with it.
Qt	Boost graph library	Team know the basics of Qt and C++. Good support. Cross platform.	Poor tools for UI development.
Electron	For 2D graph : Sigma JS / D3.js For 3D graph : ThreeJS / BabylonJS(3D)	Fancy gui. Cross platform. Library dedicated to graph node.	Most team need to learn Js / Node.js from scratch. New technology.
Others Frameworks using C++ (Nana / wxwidgets)			As we already use Qt and they're similar.

Screenshot of github page with our application logo

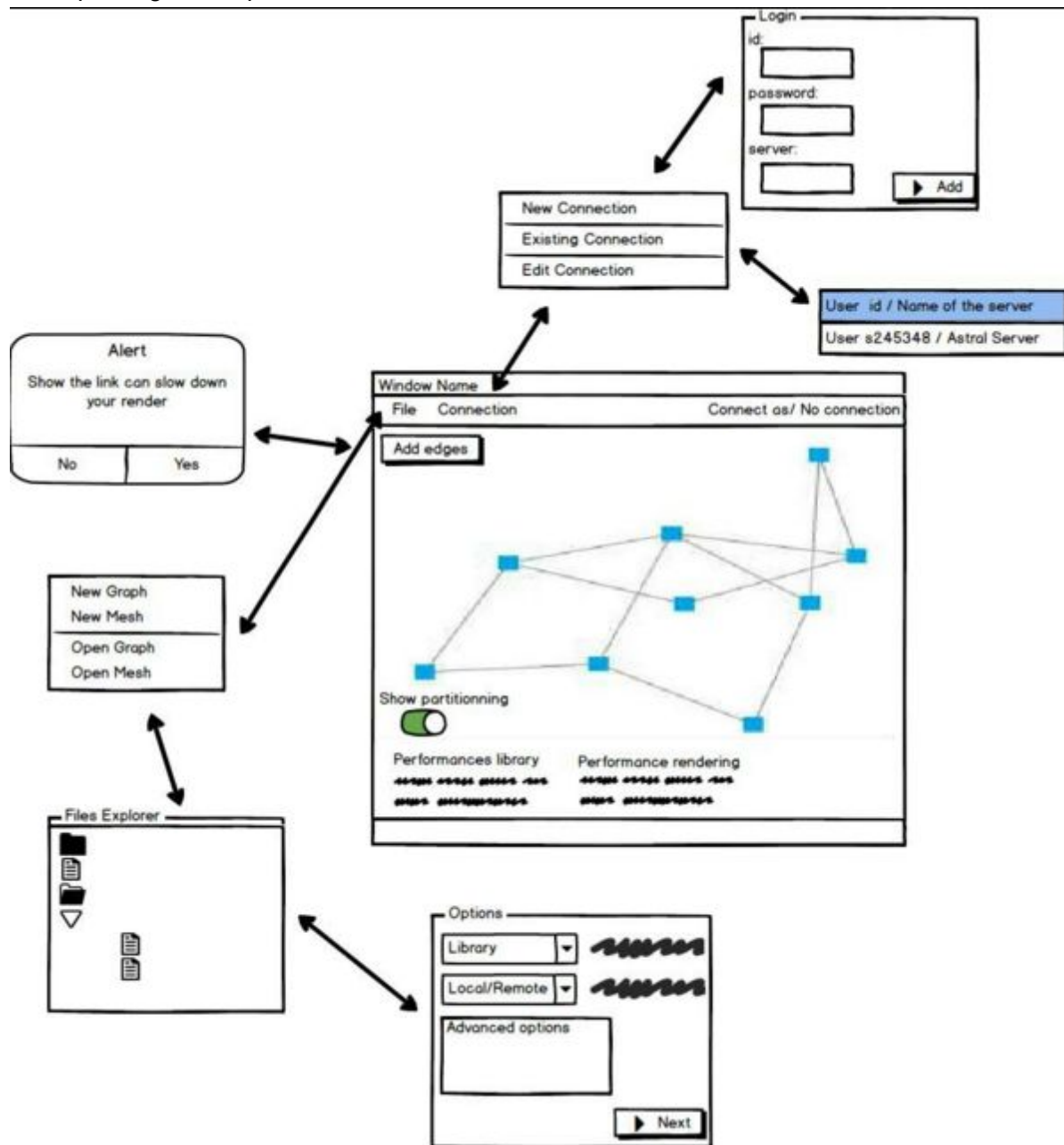
The screenshot shows the GitHub repository page for 'bnogaret / graph-partitioning'. The repository has 208 commits, 1 branch, 0 releases, and 3 contributors. The latest commit is 'bnogaret Improve README file: add images' from 6 minutes ago. The repository includes files like 'design', 'libs', 'src', 'static', '.eslintignore', '.eslintrc', '.gitignore', 'README.md', 'User_Params', and 'package.json'. The README.md file is displayed, featuring the 'PARTIFY' logo, which consists of a network graph with red and blue nodes and edges, followed by the word 'PARTIFY' in large, bold, grey letters.

How we progress working on the project(commits to days)

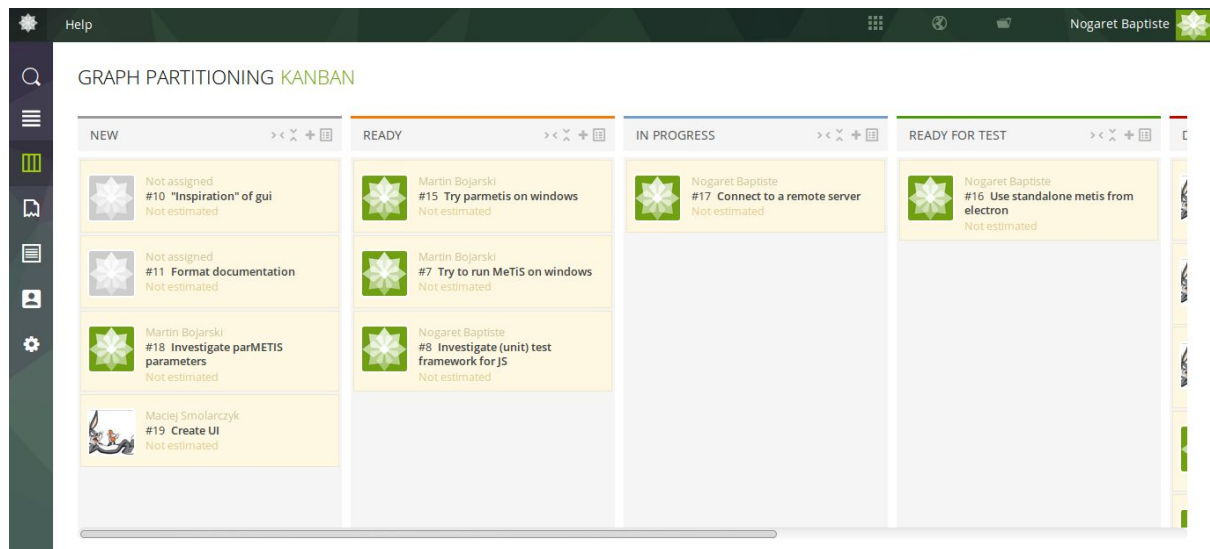


Mockup of our application

To have better understanding of our goals at the beginning we decided to create mockup of our UI. Thanks to that it was much easier to discuss about each necessary component and split the tasks among team members. Final version of the application is different than picture below because when we developed our first version of the UI it was not needful to spend time updating mockup.



Screenshot of the kanban page on taiga.io



Screenshot of the wiki page on taiga.io

