# Structure of the site:

* Attractive and informative home page: introduce the project, team and specific features
* Background and context: what is the system to be designed meant to do and in what context, initial problem statement, information about client, challenges to be met, what is need for the project to be a success
* Project requirements and scope: requirements, scope, user interface requirements
* Results of the research: resources, what we found out, conclusions, decisions
* Prototypes:
* User interface: UI features (using our HCI knowledge)
* Link to project version control repository (which we must all use and update regularly) NOT OPTIONAL
* Testing: initial strategies, testing must be automated
* Record or progress: meeting minutes and copies of bi-weekly reports
* Plans: plans for developing the final PoC design during term 2
* Video: pana pe 9 ianuarie dar are 6 minute

# BA BOGDANE CICA TOATE CACATURILE DE PE SITE SA FIE UPLOADATE PANA PE 12 DECEMBRIE, ADICA SI TESTELE SI TOT SI AUTOMATIZAREA

# ISSUES TO ADDRESS:

• What are the potential platforms, operating systems, operational environments needed. The platform is the hardware or environment needed to run support the PoC. o What did you look at? Compare the alternatives. o What did you choose and why? What examples did you find? o What are the trade-offs? • What are the potential programming languages, libraries and other software components identified. o What choices were there and what are their strengths and weaknesses? o What did you choose and why? • A description of the tool chain needed to build, setup and configure the platform, OS, settings and application. o Again, identify and compare the choices. o For example, configuration scripts, build process, version control and deployment process. o The goal is to automate as much as possible to avoid having to build or configure manually every time something changes. Also to manage all the source code and other artefacts efficiently. • Critical discussion on what you have identified as the required performance and operational capabilities for the system. o How can they be achieved? o Why is your PoC fit for purpose? Provide the evidence. • Progress made in constructing your PoC Design prototype(s) by discussing the experiment phases o Using the experiment log data you build weekly, describe what you have created so far, showing the design and other features. Especially note when things change. Use appropriate language, notation (e.g., UML) and terminology to describe the design. Justification of choices and trade-offs, especially for use of design patterns to refactor, optimise and integrate your solution phases. o You might cover one or more of the following: How you are configuring your OS, what packages are included or removed. Discussion on the build and integration processes. Scripts or other processes for configuring and launching the OS and application. The layers identified (e.g., hardware, OS, library/framework, application) and how they communicate with each other. The public APIs that your platform and/or OS make available to allow the application to be written. Identification of what has been adapted from the open source community and what is to be newly developed or configured. • Your research, what you investigated, why it is relevant, what did you discover. • How is the work good Computer Science? • Testing Strategy o Investigation of relevant available testing tools and methods. o Comparison of alternatives. o How automation can be achieved. o Results of experiments on the experimental phases and PoC prototype(s).