**CMS Plan Report**

**Glossary**

**1. Software Development Principles**

**2.Kanban Methodology**

**3.Summary**

**4.References**

**Software engineering principles**

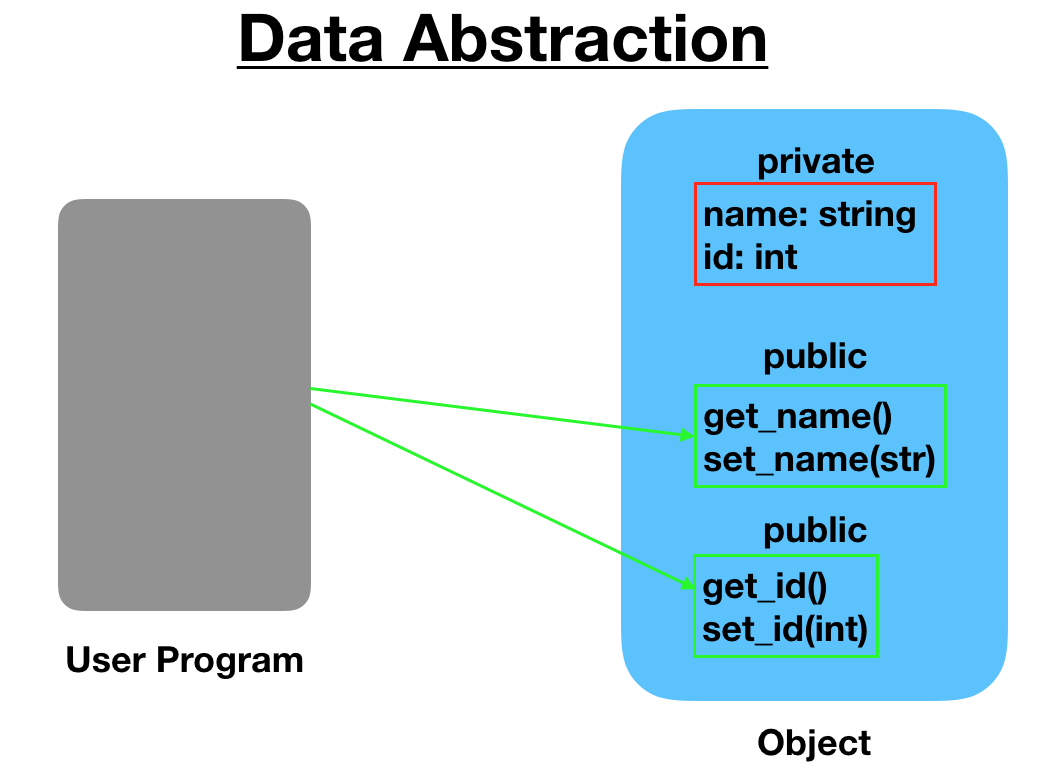
**1**

**Abstraction** is hiding complicated functionality from the eyes of the user or client; Abstraction also leaves simple to understand functionality behind for the user or client.

Applying this to CMS Abstraction helps hide unnecessary functionality that may confuse the stakeholders or overwhelm them with major details.

In turn it will leave the less complicated functions alone which would be the log in form, registration, class links or materials, messaging or forum discussions.

Abstraction is a simple principle to follow as all it requires is isolating how the is structured CMS, there is little chance for error through isolation when making changes to its structure.



**Example of data abstraction**

**Source:** Ozdil, M.H. (2021). *What is ‘Abstraction’? What are the differences between Abstract Classes and Interfaces?* [online] Let’s Do It PL. Available at: <https://medium.com/lets-do-it-pl/what-is-abstraction-what-are-the-differences-between-abstract-classes-and-interfaces-383dc9347282>.

**Software engineering principles**

**1**

**(Continuation)**

**Reusability** is a principle that focuses on repeated usage of code or functions, every time a user logs in and logs out a website reuses the same function that allows the user to log in and out.

Reusability is also used when ordering from Amazon for example, whenever a user orders something they press buy now they are navigated to fill in a form that requests their details, so the delivery knows where to go and if the user wishes to do so the second time it repeats the same process.

In CMS the process in this case is repeated through log in to the CMS, whenever the student or lecturer log in the CMS reuses this functionality repeatedly.

Reusability is cost effective as it allows companies and businesses to save time and money in the long term since the functionality is reused and no need to change it.

Reusability also prevents bloating of code meaning that the functions are not slow or consume resources at a large amount and it is to make sure that the functions don’t affect the output that may result from that function.

**Software engineering principles**

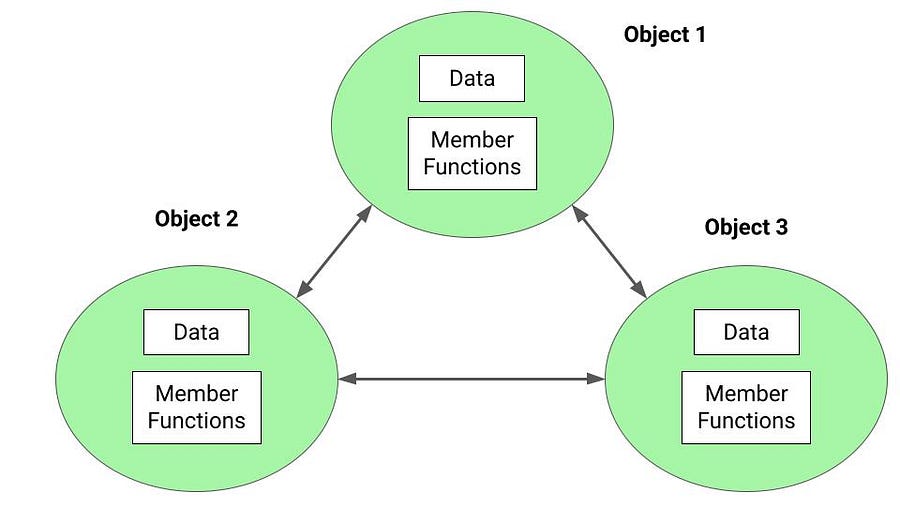
**1**

**(Continuation)**

**Encapsulation** hides internal operations in a system, this decreases any security breaches that may occur within a system, Encapsulation like Abstraction hides values but not exactly like Abstraction.

Encapsulation mostly focuses on Security concerns of the system so in CMS Encapsulation would hide sensitive information about the students, lecturers and any employees that are in the college.

Encapsulation boasts flexibility as it allows the developer to have values or classes involved in the CMS set to read or write only, write makes changes to those values and classes and read displays the desired results.



**Example of Encapsulation**

**Source:** Nishad, A. (n.d.). *Encapsulation in Object-Oriented Programming*. [online] [www.enjoyalgorithms.com](http://www.enjoyalgorithms.com). Available at: <https://www.enjoyalgorithms.com/blog/encapsulation-in-oops>.

**Kanban Methodology**

**2**

**What is Kanban?**

The methodology was first developed by a Japanese engineer called Taiichi Ohno during the 1940s.

**Kanban** is a framework that is popular amongst various companies and development teams that lets managers and team leaders keep an eye on their progress on projects or tasks they assign to employees and team members which supports improvement, flexibility and consistent workflow in task management.

Kanban's focus is primarily on project transparency and collaboration with members on a team, so teamwork and project understanding are two of the many key features of Kanban.

How it works is that the methodology visualizes a Kanban board that is compiled of what task is assigned to who, what is the goal of that task and what should it do.

Kanban board is split into 4 sectors the backlogs, in progress, review and completed.

Backlogs are requirements compiled based on what the customers or stakeholders wish to see.

In progress are tasks and requirements that are being worked on to suit the needs and requests of the customer or stakeholder.

Review is where the tasks are flagged for review to discern if the requirements are met and are clear to be implemented.

**Kanban Methodology**

**2**

**(Continuation)**

Completed is where the requirements that are met are implemented into the project and marked off as a complete request.



**Example of a Kanban board**

**Source:** Wikipedia Contributors (2023). *Kanban (development)*. [online] Wikipedia. Available at: <https://en.wikipedia.org/wiki/Kanban_%28development%29>.

**Kanban Methodology**

**2**

**(Continuation)**

**Examples of where Kanban saw huge usage**

Kanban is used constantly throughout companies and development teams, but where has it seen most usage?

We already know Kanban originated in Japan during 1940s, it first started being used in Toyota in 1940s and is still being used even today in Toyota.

This in turn supported their industry through production improvements and a constant increase in customer satisfaction.

IT industry is a major example of where Kanban is used more, Kanban helps the IT industries how best to split tasks of huge projects this determines which task is quick to finish and which is more important to finish.

This allows the IT industries to reach project scheduling and finishing on time for release of the project.

Hospitals see a lot of use for the Kanban Methodology, Kanban helps hospitals through management of processed patients so what rooms are vacant for processing patients and which patients to release from care.

Kanban helps with managing hospital storage and equipment; this allows doctors or nurses to make certain that the hospital is stocked with necessary equipment and chemical compounds for future patients.

**Kanban Methodology**

**2**

**(Continuation)**

**Why choose Kanban for the CMS?**

Designing and developing a CMS is a large undertaking and has to be done efficiently while there are other frameworks that support teamwork and collaboration, Kanban I would say is a perfect tool for the job.

Kanban can allow us to split the CMS into more manageable and understanding tasks and requirements that the stakeholders set for the development and design teams to work on.

Kanban encourages collaboration and collaboration means high morale within the development and design departments, this will ensure that the project is done quickly and efficiently.

**Kanban Methodology**

**2**

**(Continuation)**

**Kanban's Strengths and Weaknesses**

**Strengths**

-Understanding and comprehensive tasks help a team understand what they are working with as Kanban offers Kanban Cards to help describe the requirements.

-Clear and Concise progress readability show how the tasks are progressing, which tasks are not finished, which ones are finished, and which are stalled and falling behind the intended schedule.

-Team efficiency allows the time to react quicker when issues arise. This is coupled with how the tasks are visualized and described which helps spot bottlenecks quickly.

-Improved focus and visibility in tasks decrease concerns about missing out on an important detail since the team will be more focused on their assigned objectives while saving up energy on more immediate concerns.

**Kanban Methodology**

**2**

**(Continuation)**

**Weaknesses**

-Lack of scheduling and finishing dates makes it difficult to predict when the tasks will be finished or when the project itself will be finished.

-Overwhelming understanding of the tasks is one of the major downsides of Kanban, if a team is dealing with a large project especially that has a vast number of requirements it can be difficult to organize all of the requirements into a simple to understand state and may become a burden to team morale.

-Keeping up with constant updates to the board is important so as to not have outdated information or requirement, if one requirement is done but is not marked as such another team member might start working on a requirement that is already finished as a Kanban board is updated in real time as the process progresses.

**Summary**

**3**

To summarize the principles and the framework above should help finish the CMS development process quickly and on time, the Kanban methodology would be best for the CMS project because of how it is organized into small parts rather than a large scope that can prove to be complicating than understanding and implementing and utilizing the principles above to best ensure the satisfaction and safety of our lecturers and students that will use the CMS while also meeting and addressing their requirements that they have requested to see in the CMS.

**References**

**4**

**Note:Most of the references were used for the powerpoint**

Tutorialspoint.com. (2019). *UML - Deployment Diagrams*. [online] Available at: <https://www.tutorialspoint.com/uml/uml_deployment_diagram.htm>.

‌

www.smartdraw.com. (n.d.). *Component Diagrams - See Examples, Learn What They Are*. [online] Available at: <https://www.smartdraw.com/component-diagram/#:~:text=A%20component%20diagram%2C%20also%20known>.

‌

Rani, B. (2018). *Unified Modeling Language (UML) | Class Diagrams - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams/>.

‌

Pacific Research. (n.d.). *What is Iterative Development and Why Should You Use It?* [online] Available at: <https://www.pacific-research.com/iterative-product-development/>.

‌

GeeksforGeeks (2019). *System Testing - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/system-testing/>.

‌

Katalon (2023). *What Is Integration Testing And Its Types?* [online] katalon.com. Available at: <https://katalon.com/resources-center/blog/integration-testing>.

**References**

**4**

**(Continuation)**

Moradov, O. (2022). Unit Testing: Definition, Examples, and Critical Best Practices. *Bright Security*. [online] 24 May. Available at: <https://brightsec.com/blog/unit-testing/>.

‌

Testsigma Blog. (2023). *What Is Test Case? How To Write Test Cases In Software Testing?* [online] Available at: <https://testsigma.com/guides/what-is-test-case/>.

‌

Git (2019). *Git - About Version Control*. [online] Git-scm.com. Available at: <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>.

‌

BrowserStack. (n.d.). *Importance Of Code Reusability In Software Development*. [online] Available at: <https://www.browserstack.com/guide/importance-of-code-reusability>.

‌

Coursera. (2023). *What Is Encapsulation?* [online] Available at: <https://www.coursera.org/articles/encapsulation>.

‌

Anon, (2023). *Key Principles of Software Architectures - mokunote*. [online] Available at: <https://mokunote.com/principles-of-software-architectures/> [Accessed 31 Mar. 2024].

‌Virtana. (n.d.). *What is horizontal scaling (scaling out)?* [online] Available at: <https://www.virtana.com/glossary/what-is-horizontal-scaling/#:~:text=Horizontal%20scaling%20is%20the%20ability>.

‌ **References**

**4**

**(Continuation)**

Amazon Web Services, Inc. (n.d.). *What is Load Balancing? - Load Balancing Algorithm Explained - AWS*. [online] Available at: <https://aws.amazon.com/what-is/load-balancing/#:~:text=Load%20balancing%20is%20the%20method>.

‌

Sisense. (2017). *14 Useful Methods for a Successful Data Visualization (With Examples)*. [online] Available at: <https://www.sisense.com/blog/10-useful-ways-visualize-data-examples/>.

‌

www.intertrust.com. (2021). *What is data interoperability? – Intertrust Technologies*. [online] Available at: <https://www.intertrust.com/blog/what-is-data-interoperability/>.

‌

DreamClass. (n.d.). *What is a Student Information System (SIS)?* [online] Available at: <https://www.dreamclass.io/what-is-a-student-information-system-sis/>.

‌

Qlik. (n.d.). *What is Data Modeling?* [online] Available at: <https://www.qlik.com/us/data-modeling>.

‌

Brush, K. and Kirvan, P. (2019). *What is a Learning Management System (LMS) and What is it Used For?* [online] SearchCIO. Available at: <https://www.techtarget.com/searchcio/definition/learning-management-system>.

‌

**References**

**4**

**(Continuation)**

Vice Vicente (2021). *What Is a Security Audit? The Basics You Need to Get Started*. [online] AuditBoard. Available at: <https://www.auditboard.com/blog/what-is-security-audit/>.

‌

Radigan, D. (n.d.). *What is kanban?* [online] Atlassian. Available at: <https://www.atlassian.com/agile/kanban>.

www.momentup.com. (n.d.). *What is Agile Scrum Methodology and How Can it Help Your Startup? | Momentup*. [online] Available at: <https://www.momentup.com/blog/what-is-agile-scrum-methodology-and-how-can-it-help-your-startup> [Accessed 31 Mar. 2024].

‌

Raymond, D. (2023). *Kanban Boards: Top 10 Cons, Disadvantages & Limitations*. [online] Articles for Project Managers - The Project Management Network. Available at: <https://projectmanagers.net/kanban-boards-top-10-cons-disadvantages-limitations/>.

‌

Edrawsoft. (n.d.). *Communication Diagram Explained | EdrawMax Online*. [online] Available at: <https://www.edrawmax.com/article/communication-diagram-uml.html>.

GeeksforGeeks (2017). *Unified Modeling Language (UML) | Sequence Diagrams - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>.

‌

**References**

**4**

**(Continuation)**

FONSECA, L. (2022). *How to Make a Class Diagram [+Examples]*. [online] Venngage. Available at: <https://venngage.com/blog/class-diagram/>.

WhatIs.com. (n.d.). *What is caching? - Definition from WhatIs.com*. [online] Available at: <https://www.techtarget.com/whatis/definition/caching>.

Boaron, A. (2021). *Kanban Zone*. [online] Kanban Zone - Software, Method, and Expertise to efficiently achieve all your goals. Available at: <https://kanbanzone.com/2021/industries-where-kanban-method-can-be-used/>.

**Image References**

**Abstraction**

Ozdil, M.H. (2021). *What is ‘Abstraction’? What are the differences between Abstract Classes and Interfaces?* [online] Let’s Do It PL. Available at: <https://medium.com/lets-do-it-pl/what-is-abstraction-what-are-the-differences-between-abstract-classes-and-interfaces-383dc9347282>.

‌

**Encapsulation**

Nishad, A. (n.d.). *Encapsulation in Object-Oriented Programming*. [online] www.enjoyalgorithms.com. Available at: <https://www.enjoyalgorithms.com/blog/encapsulation-in-oops>.

‌

**Kanban**

Wikipedia Contributors (2023). *Kanban (development)*. [online] Wikipedia. Available at: <https://en.wikipedia.org/wiki/Kanban_%28development%29>.