

Project Management

For Res. & Dev. Methods

James MG Williams

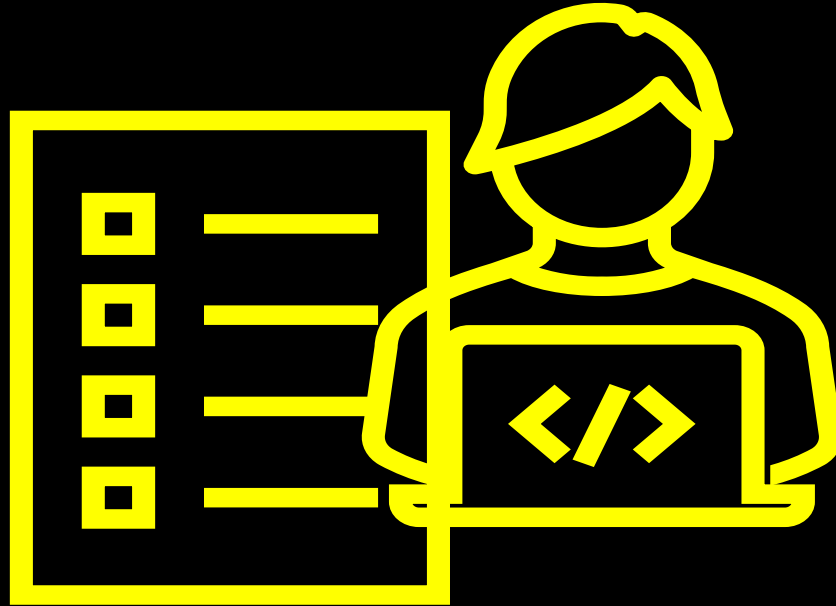




Introduction & Overview

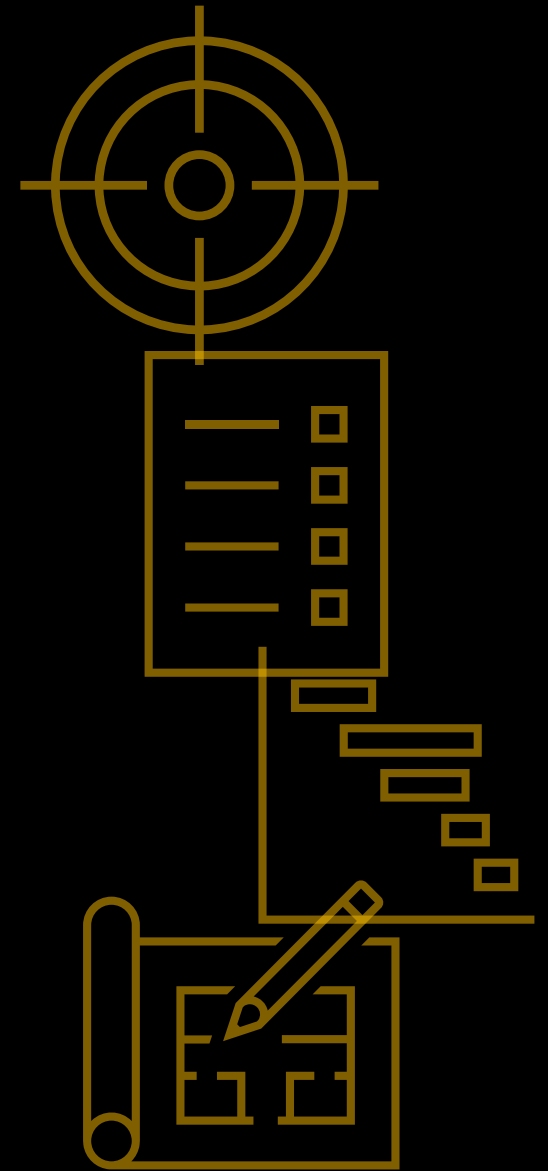
1. Project Management
 - Key considerations & Stages
2. Project Planning
 - inc. Project Scope
 - Inc. Proposed solution & resources
3. Project Planning & Management tools
4. Project Development Methods
5. Project Design
6. Project Manager & Risk Management

Introduction: Project Management



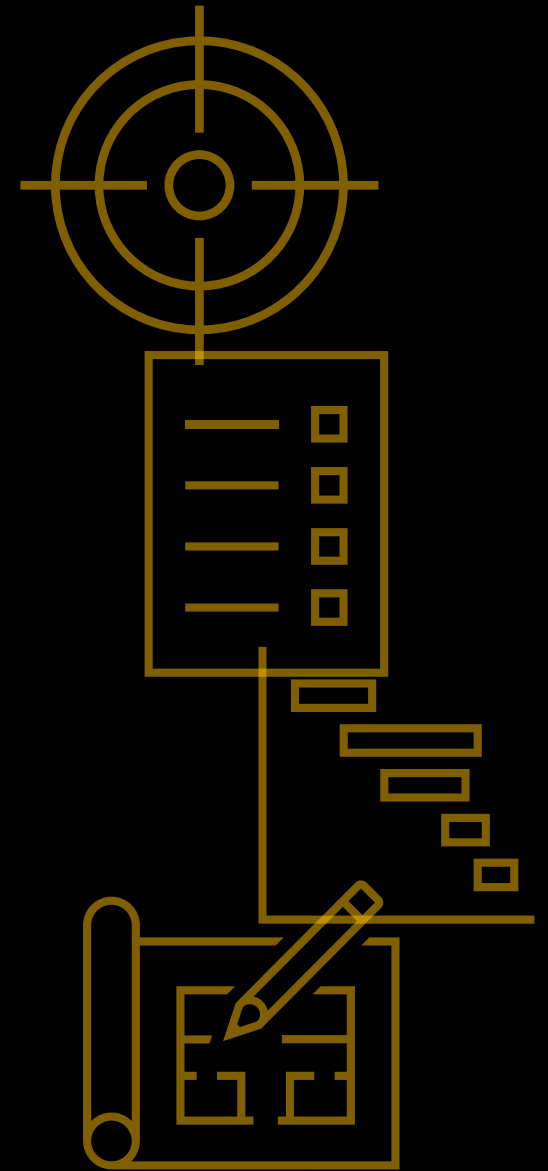
Project Management

- **According to the Project Management Institute PMI,** Project management is the discipline of planning, organizing, motivating, and controlling resources to achieve specific goals. (1)
- A project is a temporary endeavour designed to either solve a problem, create something or produce a positive result. (1)
- **Purpose & Positive Social Impact (PSI):** Typically, a project is created to bring about beneficial change and add value to society



Project management skills

- While project management skills are obviously important for project managers, interestingly **the methods and tools that project managers use can be helpful for everyone.**
- For example: Completing Assignments (An Assignment can be equated with a Specification, and you are the Project Manager).



4 elements of Project Management

All projects have 4 common elements:

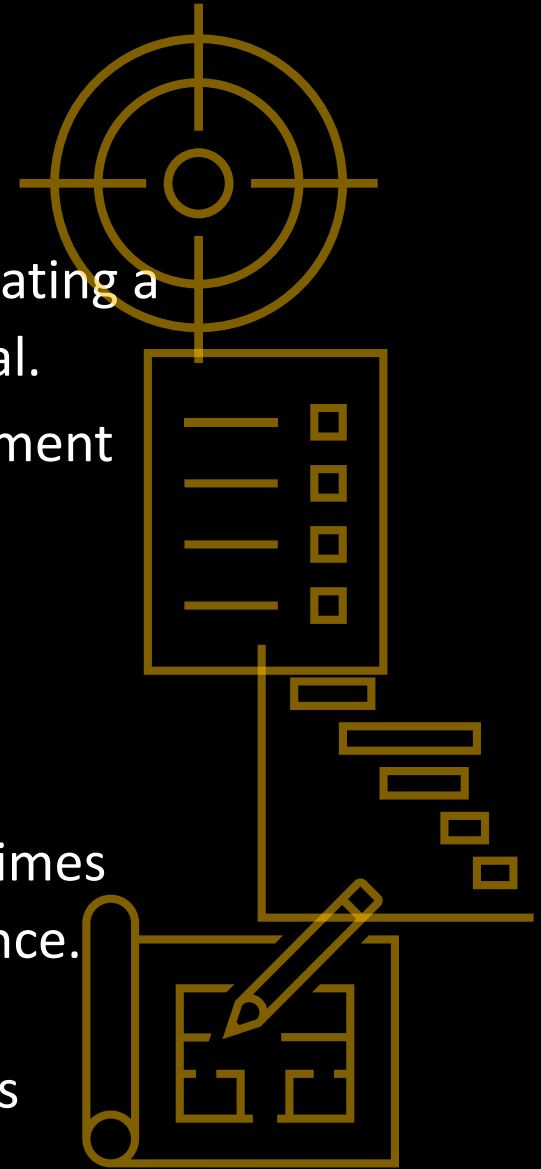
1. A **Starting** point
2. A **desired outcome**
3. A **procedure/process** for getting to the outcome
4. An **End** point.



Project Stages

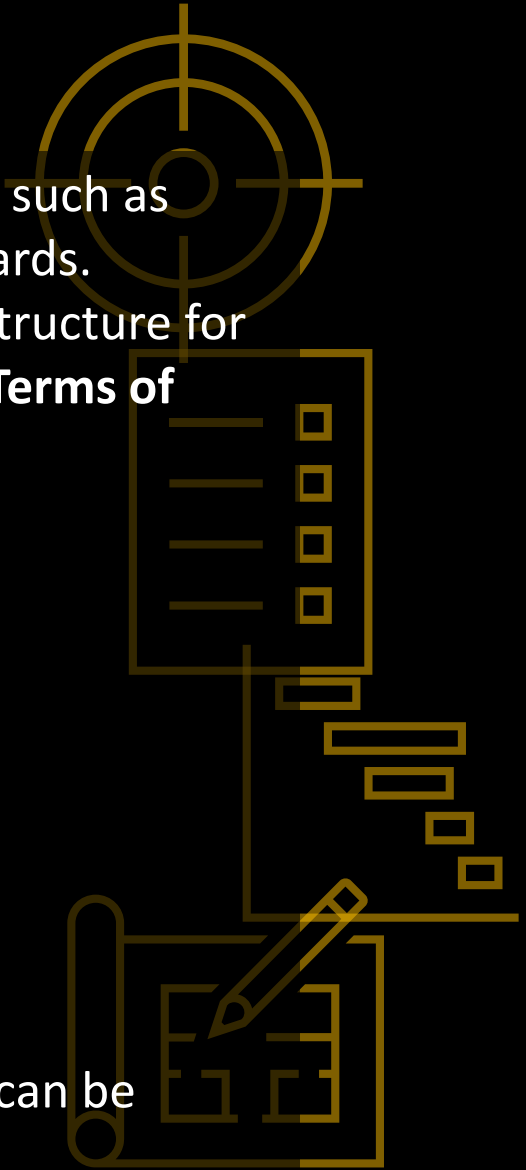
A project generally has the following stages:

1. **Defining a project:** Ideation/idea, coming up with the initial concept, creating a Project specification / Project initiation document (PID) / Project proposal.
 2. **Planning** a project includes creating a project plan and schedule to document key stages.
 3. **Monitoring** a project over its lifecycle.
 4. **Implementing** a project within agreed procedures and to specification.
 5. **Testing & Evaluation** against initial specification.
 6. **Reporting** a project outcome involves writing a formal report and sometimes presenting the results e.g. To the Project Sponsor, Trade Shows, Conference.
- **Note:** Throughout a project people develop many Employability skills such as creativity and management including planning and communication skills.



Bg. Project Spec. / Specification

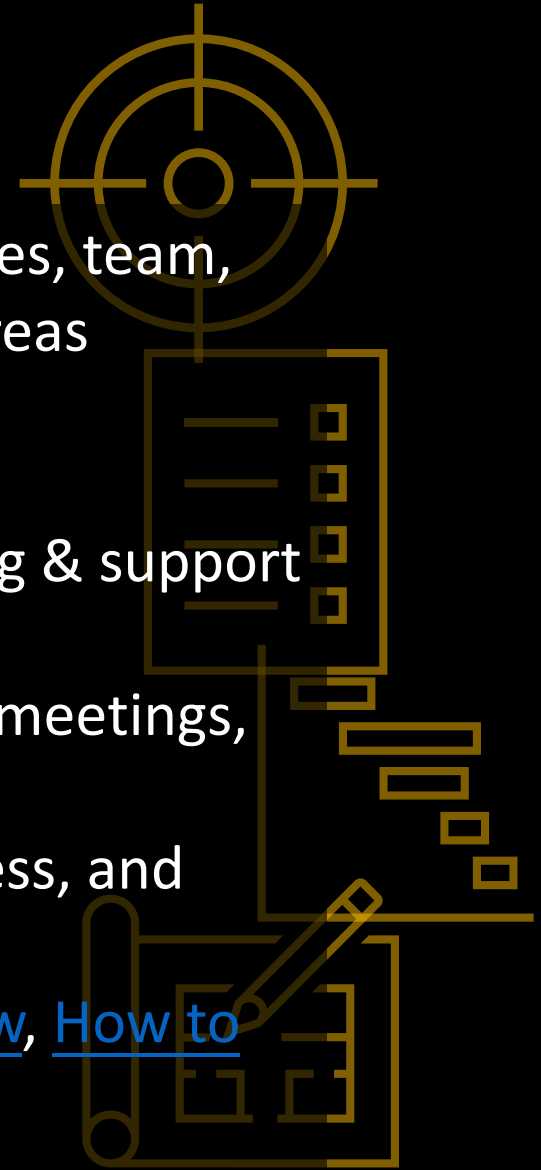
- Preparing a **project specification** requires consideration of several important factors such as **requirements, timescale, resources, budget, risk** and conformance with legal standards.
- The acronym **BOSCARDET** (BOS-CAR-DET) provides a useful example acronym and structure for a '**Project specification**' sometimes referred to as a '**Project Charter**' or a Project's '**Terms of Reference**', headings/sections include:
 - ▶ Background
 - ▶ Objectives
 - ▶ Scope
 - ▶ Constraints (inc. budgets)
 - ▶ Assumptions
 - ▶ Reporting
 - ▶ Dependencies
 - ▶ Estimates (inc. costs)
 - ▶ Timescales.
- This structure contains no specific heading for costs/budgets - these considerations can be included within 'Constraints' or 'Estimates'
- Since projects vary considerably there is no standard universal structure for a Project document.



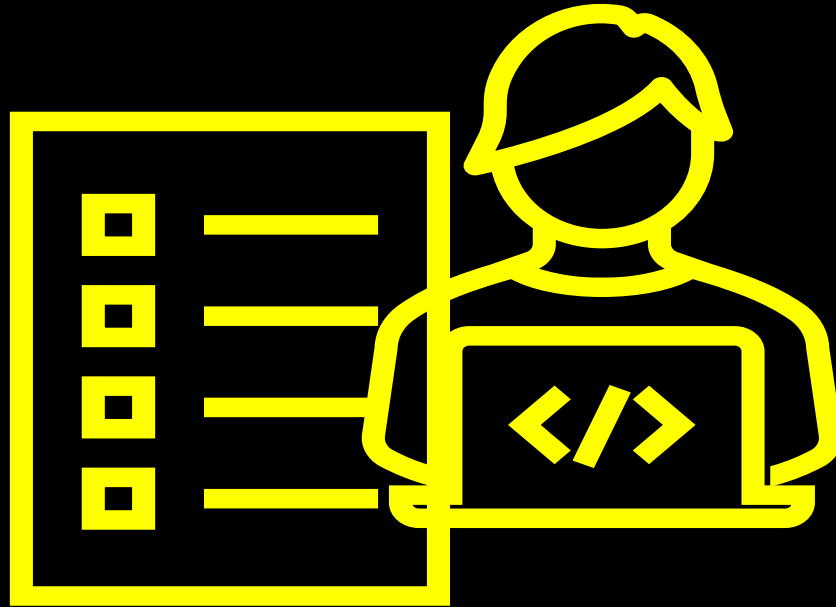
Bg. Project specification & PID:

A **Project Spec. / Initiation Document (PID)** will include project objectives, team, project team roles and communication plans. For example:, some key areas include:

- ▶ **Describe Project purpose** inc. aim, objectives and deliverables.
- ▶ **State parameters** inc. timescales, budgets, scope, authority (reporting & support structure).
- ▶ **State people involved** and the way the team will work (frequency of meetings, decision-making process).
- ▶ **Establish 'review-points'** inc. when to meet, review and check progress, and how progress and results will be measured.
- **For more information about a PID document, click here:** [PID Overview](#), [How to make a PID document](#), [Key PID considerations](#), [Example PID](#).



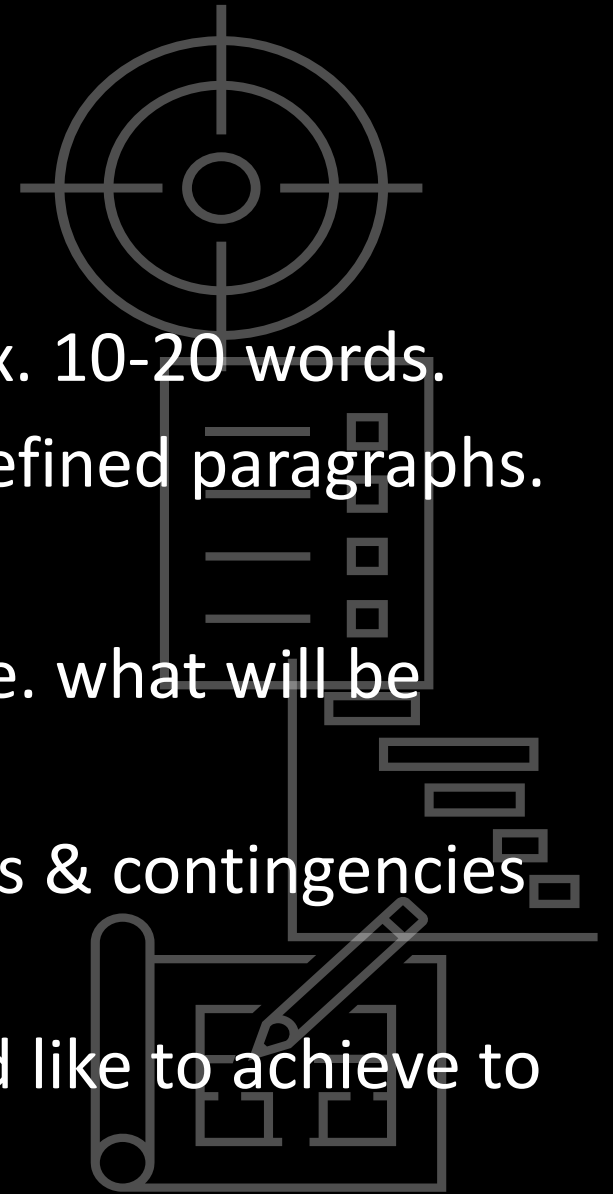
Project Planning



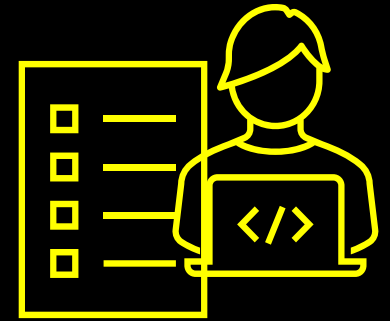
Typical Project Plan

A typical project plan, includes:

- **Project idea/Aim:** One simple clear statement, approx. 10-20 words.
- **Project description:** A brief description, approx. 1-3 refined paragraphs.
- **Project objectives:** Several clear SMART objectives.
- **Project Scope:** intended **deliverables** of the project i.e. what will be produced by the project, and the **timeframe**.
- **Project Constraints & Risks:** including limitations, risks & contingencies and resources.
- **Proposed solution:** states the outcome that we would like to achieve to the problem identified.



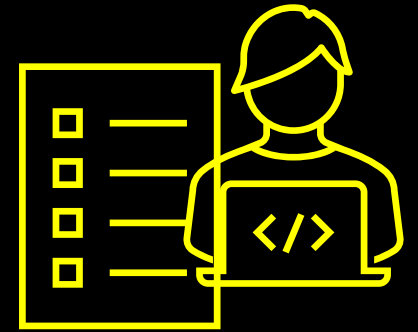
Project Scope



- A project's **scope** sets a project's **boundaries** in terms of e.g. time, resources and deliverables.
- You need to consider the **project's constraints** such as time, skillset, experience & resources and decide what can be realistically achieved.
- You can initially set the bar a bit higher and reign-it-in if needed as the project progresses or towards the end.
- Your requirements can be modified accordingly.
- Project scope outlines the project's tasks, activities, timeline, and deliverables.

A Project's Scope also includes the project's **key stages (aka milestones)**, **stage completion dates** and **intended outcomes**.

Project Scope



For your project, you can define the **Project Scope** by simply including a:

Project timeline: eg. Sept. 2024 to April 2025.

- With Key Project stages & tasks and approximate completion dates for each stage and task.

Project outcomes: Include a brief description of what you intend to create/build (e.g. an App./Game) and any other key outputs e.g. best practice report. You can state the project scope will be documented further in the Project Gantt chart and Requirements specification.

Typical Chapters/stages of a Project:

To create your Project scope, including a **schedule and timeline** for the project, consider the following typical chapters of an Academic Computing project report. These can be used to create key stages of a project and included in a project schedule. To create a 'Project schedule' you can create a **list/table** with **task items** such as those below and **place completion dates** alongside each one.

1. **Introduction** (inc. Project Aim & Objectives).
2. **Literature review** (Secondary Qualitative research).
3. **Research Methodology**
4. **Requirements Analysis & Design**
5. **Practical Implementation / Development**
6. **Primary research analysis** (e.g. User Testing, Survey Questionnaire, Interviews) >
Often includes Primary Quantitative & Qualitative data analysis.
7. **Evaluation** (Analysis of all data identifying correlations and links to the project Aim & objectives).
8. **Conclusions & Recommendations.**
9. **Reflection.**



Example: Academic PROJECT DEADLINES

Documenting key dates will help you to create a **Project schedule & Gantt chart**.

Project Proposal tutorials:	Induction week, end of Sept. and October first 2 weeks.
Project Proposal:	By Monday 18th October: Project proposal and Ethics form to be submitted.
Interim Viva:	During Wed 19th/ Thurs 20th January: Presentation to include Project introduction, Literature review, Analysis of secondary findings, progress on Research Methodology and any preliminary design/implementation work.
Final Report:	By Thursday 28th April. To include all chapters from previous slide.
Final Viva:	During Wed 4th/Thurs 5th/Fri 6th May – timeslots are allocated.
Project Poster:	By Friday 6th May.
Degree Show Exhibition: Project Poster & Demonstration.	On Friday 6th May.

Bg. Example schedule:

Task	Starting Date/End Date
Complete the project proposal and ethics form.	01/10/2022 – 21/10/2022
Start Project Report and Literature review.	22/10/2022 – 01/11/2022
Complete Literature review.	1/11/2022 – 14/11/2022
Create Design work and initial prototype.	14/11/2022 – 21/11/2022
Complete Research & Dev. Methodology	21/11/2022 – 28/11/2022
Begin Development of website	01/12/2022 – 21/12/2022
Create Interim Viva slides	04/01/2023 – 14/01/2023
Interim Viva	16/01/2023 – 18/01/2023
Continued to develop the website and update report a long side development.	18/01/2023 – 28/02/2023
Finish development.	28/02/2023 – 31/03/2023
Get feedback.	01/04/2023 – 07/04/2023
Finish report.	08/04/2023 – 26/04/2023
Final Viva.	08/05/2023 – 10/05/2023
Exhibition materials.	20/03/2023 – 05/05/2023

Bg. Project Management & Project Scope.

- For more information on the **key principles of Project Management** please [click-here](#) and [here](#).
- For further information on **Project Scope considerations** [click-here](#).
- **Example Project Scope statement** [click-here](#). Scope statements vary greatly between different Organisations. For further examples please [click-here](#).

S M A R T



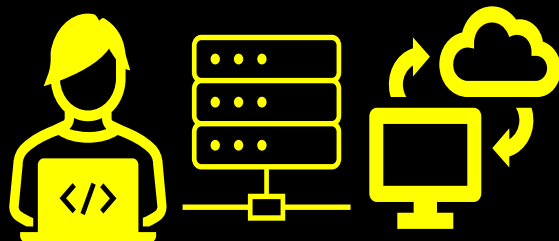
Proposed solution & Resources required

For the **Proposed solution**, document:

- What you would like to create/build or test by the end of the project.
- What outcome you would like to achieve to the problem or research question identified.

For the **Required resources**, document:

- The Hardware, Software and other resources that will be required to create/develop/implement your proposed solution.



Part 2: Project Planning & Management

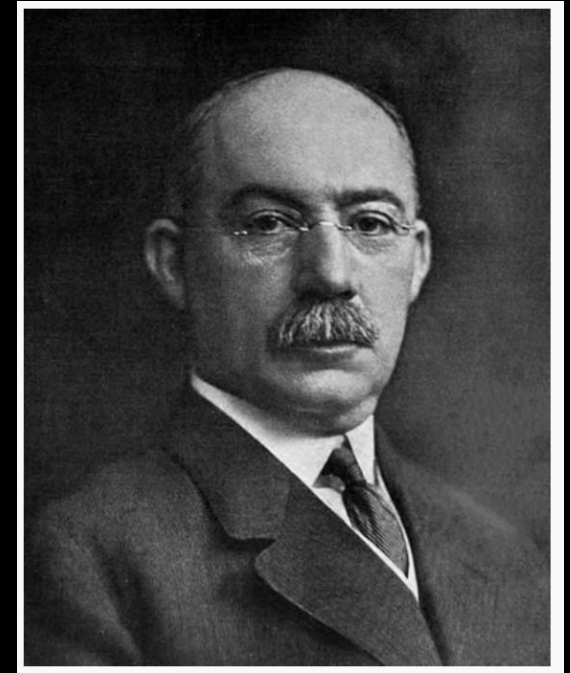
Task Name	Q1 2019			Q2 2019		Q3 2019
	Jan 19	Feb 19	Mar 19	Apr 19	Jun 19	Jul 19
Planning						
Research						
Design						
Implementation						
Follow up						

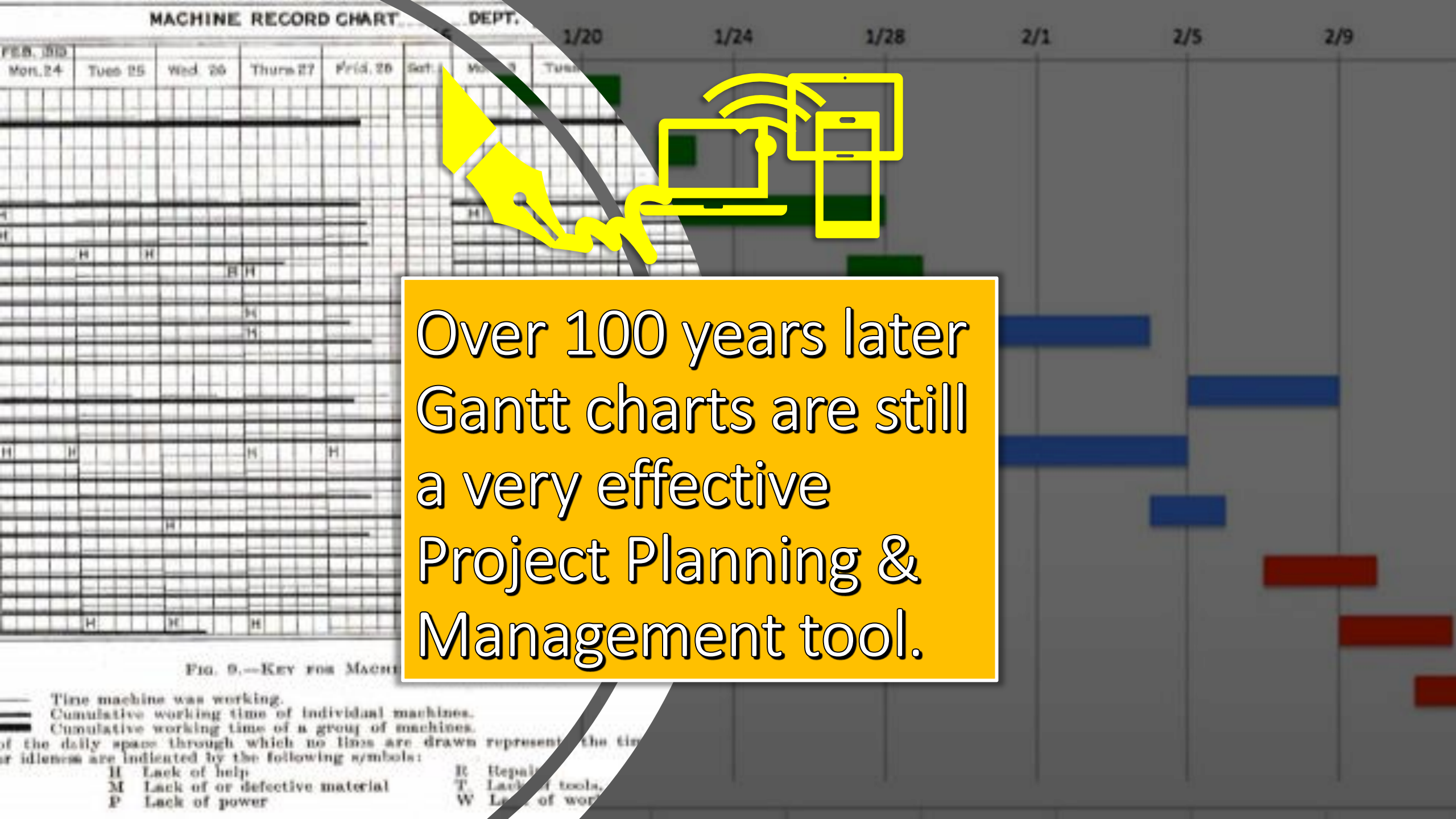
[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)

Gantt Charts

- The Gantt Chart is named after **US engineer** and consultant **Henry Gantt** (1861-1919) who created the technique in the 1910s.
- He created this **planning method** to ensure good usage of staff and machine time at **weaving mills** around the time of WW1.
- A Gantt chart shows the **various activities** which contribute to a project. The **activities/tasks** are set out against a **timeline** and the duration of **each task is visually depicted** as a **block** making it easy to see the duration of each activity.
- It is a very well-established method and is widely used in many industries today.
- Modern technology now allows for the use of Gantt charts in highly collaborative and interactive ways, making it a very efficient and popular **Project planning & management tool**.

Bg. For more information on [Gantt charts please click here](#).





Over 100 years later
Gantt charts are still
a very effective
Project Planning &
Management tool.

FIG. 9.—Key for Machine Record Chart.

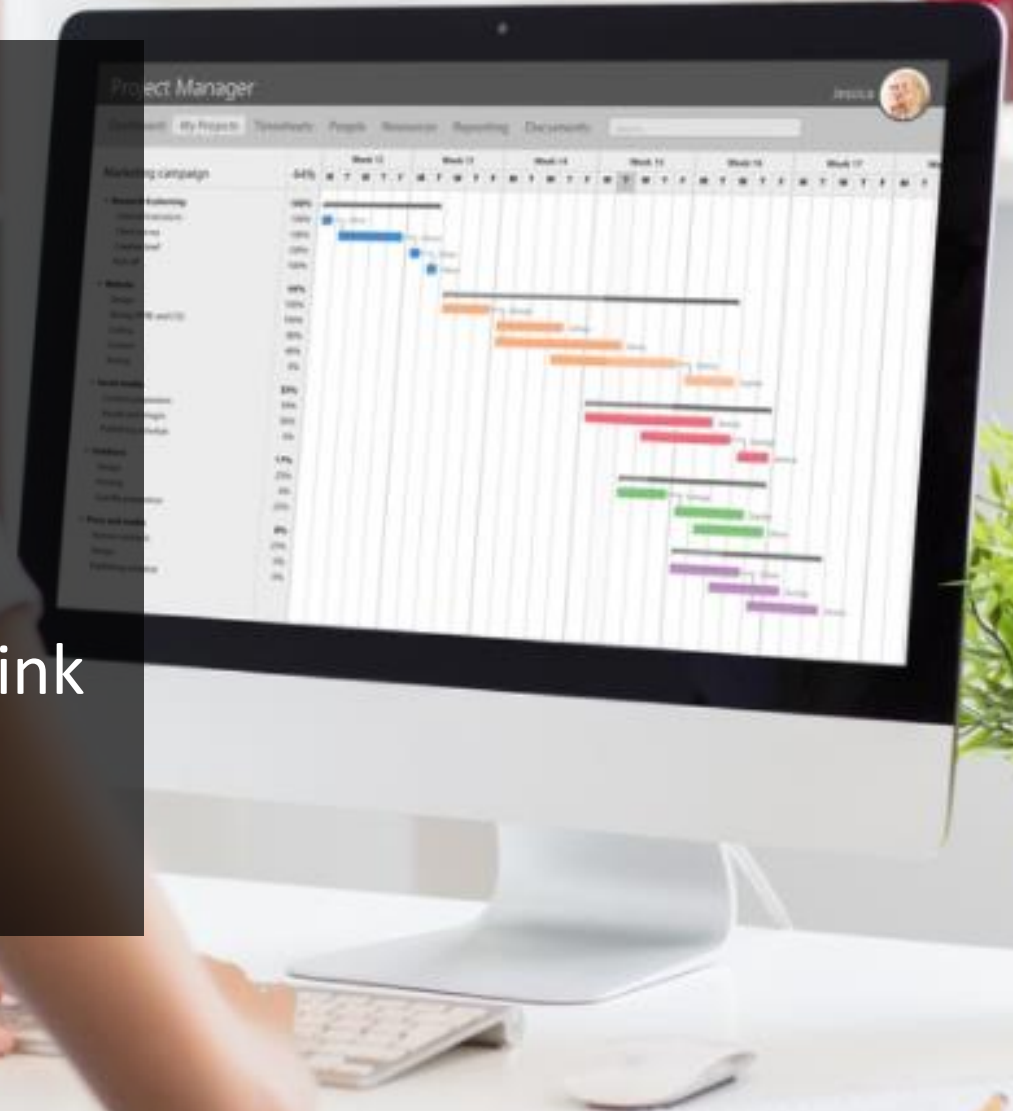
— Time machine was working.
— Cumulative working time of individual machines.
— Cumulative working time of a group of machines.
of the daily space through which no lines are drawn represent the time
or idleness are indicated by the following symbols:

H Lack of help	R Repair
M Lack of or defective material	T Lack of tools
P Lack of power	W Lack of work

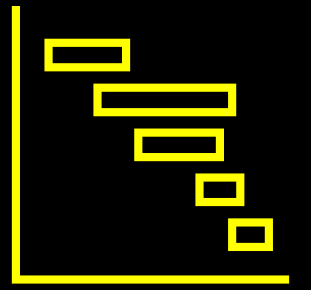
Benefits of Gantt Charts

On a Gantt chart you can easily see:

- The start date of the project.
- What the project tasks are.
- Who is working on each task.
- When tasks start and finish.
- How long each task will take.
- How tasks group together, overlap and link with each other.
- The finish date of the project



Creating a Gantt Chart: Example How to...



- Many Gantt charting tools will allow you to complete the following process in an easy and often automated way. Here is an overview of how to create a Gantt chart.
- Start the process by using a tool of your choice e.g. you could start by creating a list in word/excel, or directly with an **Online Gantt chart tool** (recommended). You can create a Gantt chart in excel by using a Bar graph and converting it into an Excel Gantt chart (a stacked bar chart). Although we recommend using a specialised Gantt charting tool to make this process easier.

1.Document a Project Start and End date.

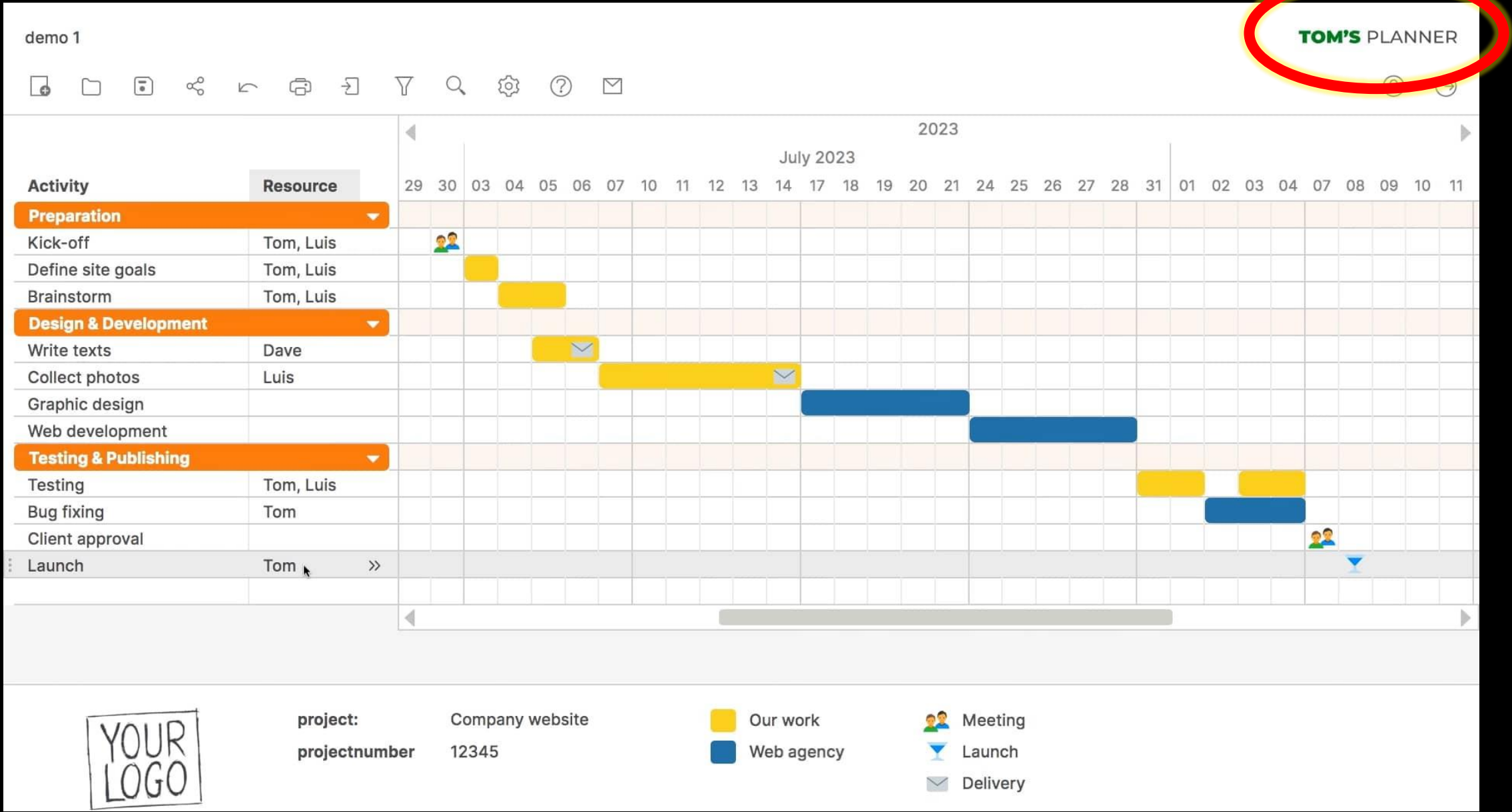
2.List your project's tasks on a separate line/row.

3.Next add start and finish dates for each task. This will set the duration and completion date for each task. This will create a **visual timeline/duration block** for each task.

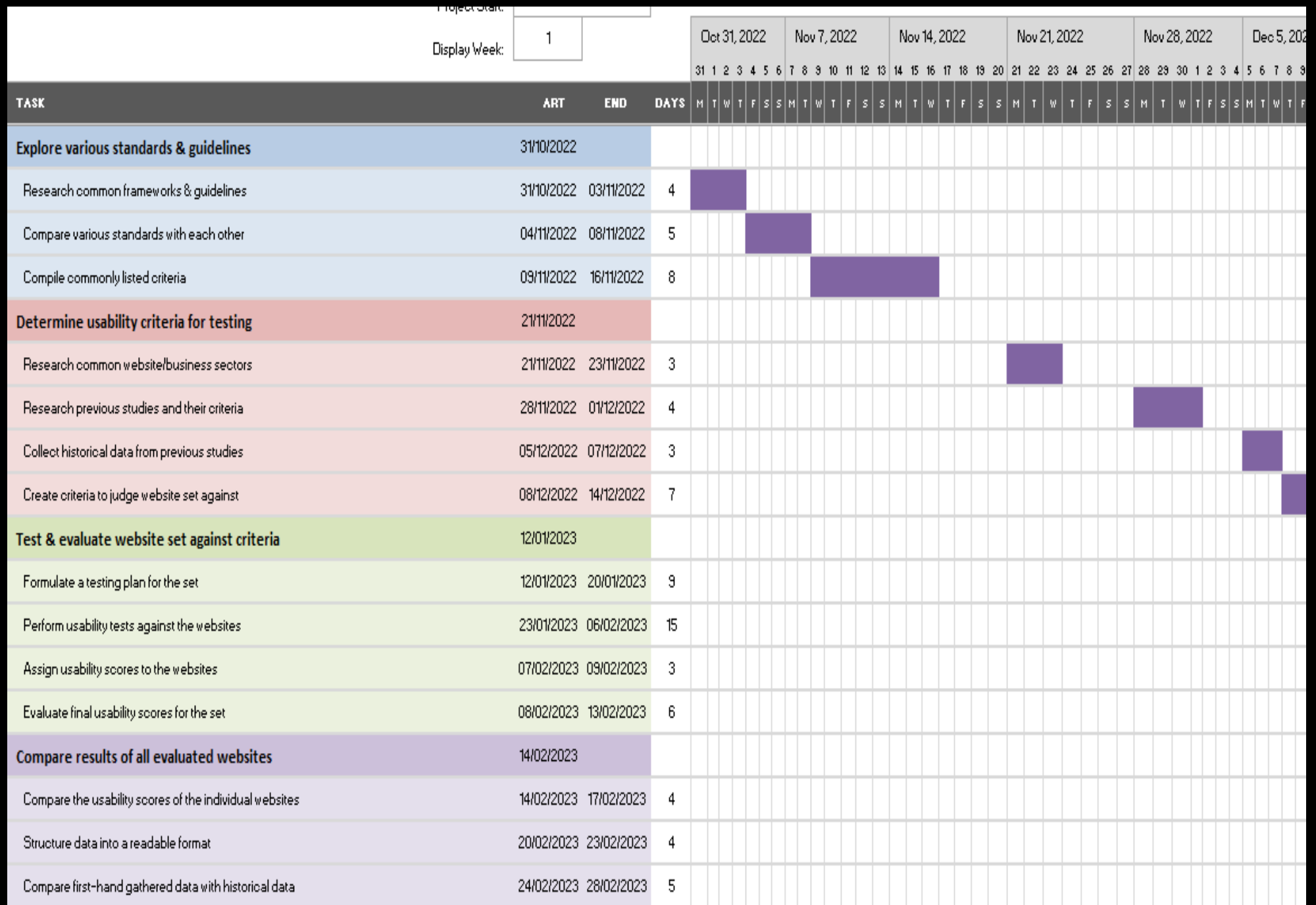
4.You can then add sub-tasks and task descriptions.

- **Bg. Group work / Project Teams:** If the project is Team-based, the project manager can then assign tasks to individual team members and invite others to collaborate on the chart.

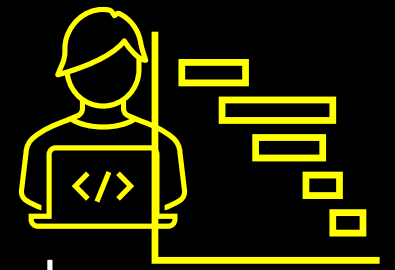
Example Gantt Chart:



Bg. Example Gannt chart:

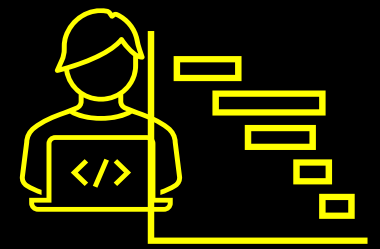


Technology: Gantt Chart Apps

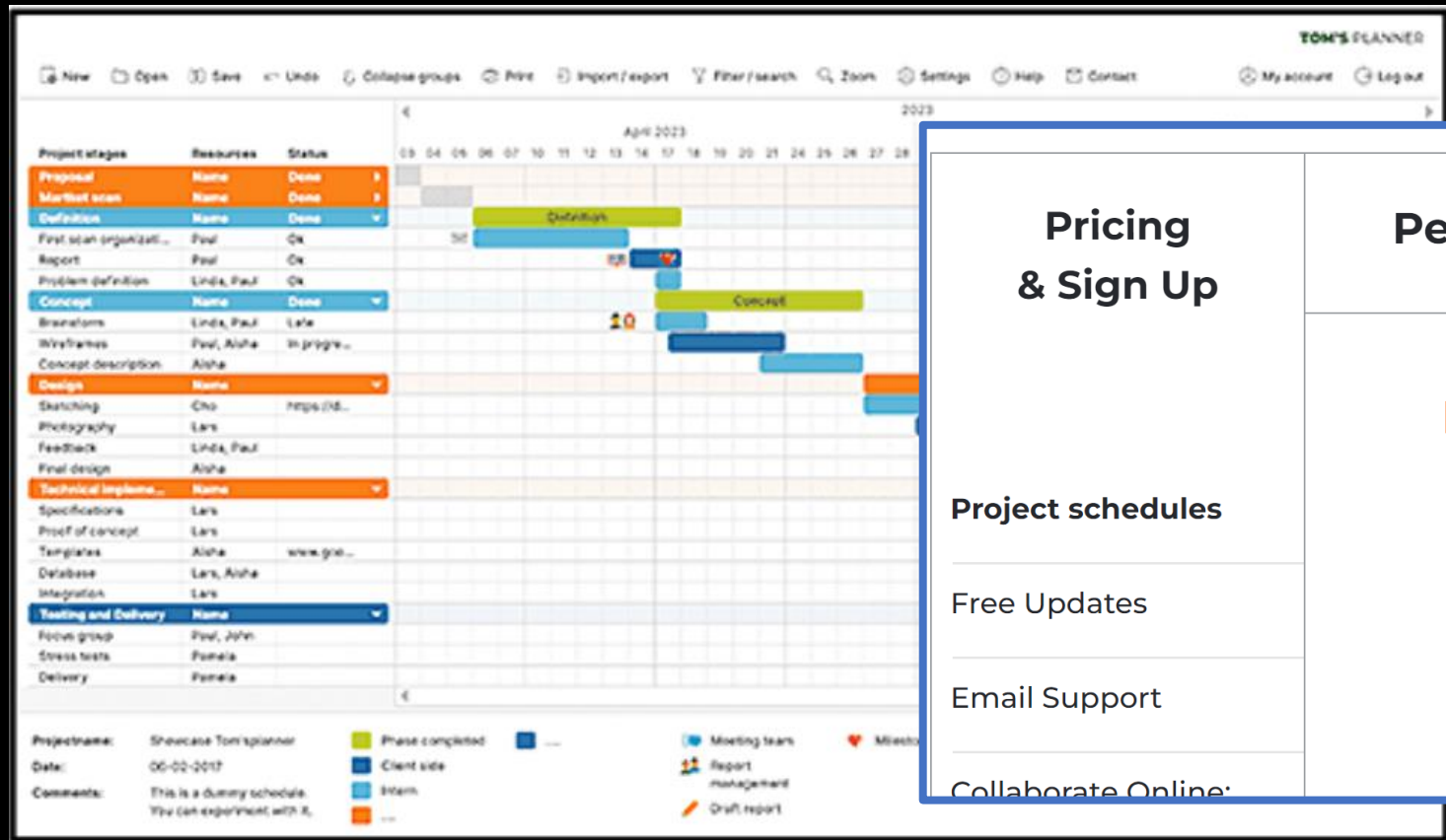


- **Microsoft** provide many tools to create a Project Timeline/Gantt chart such as [Microsoft Excel](#), [Microsoft Project](#) or [Microsoft Planner](#).
 - **Bg.** Note 1: A Gantt chart can also be created with [Microsoft Visio](#) (a Design chart App.) but this is not recommended, as it is convoluted.
 - **Bg.** Note 2: MS Apps such as MS Planner and MS Project can be integrated into MS Teams for collaborative project work.
- Google Drive also provide tools such as **Google Sheets** that can be used to create a Project Timeline/Gantt chart.
- Today, however you can easily create an online **Gantt chart** using specialised cloud Gantt Apps. Such as [Tomsplanner.com](#), [Asana.com](#), [Wrike.com](#), [Monday.com](#), [Miro.com](#).
 - Others: [TeamGantt](#), [SmartSheet](#), [Zoho](#), [Ganttter](#), [Ganttlic](#), [Creatly](#).
 - **Bg.** Note: These online Gantt chart Apps are very easy to use. If needed, you can find many easy-to-follow tutorials online that show you how to create a Gantt chart for your project in minutes. Search [Google](#) or [YouTube](#) to find useful examples.

Bg. Feature: Tomsplanner.com



- [Tomsplanner.com](https://tomsplanner.com) is a web-based collaborative Gantt chart App that allows you to easily and quickly create professional Gantt charts for project planning, management and collaboration.



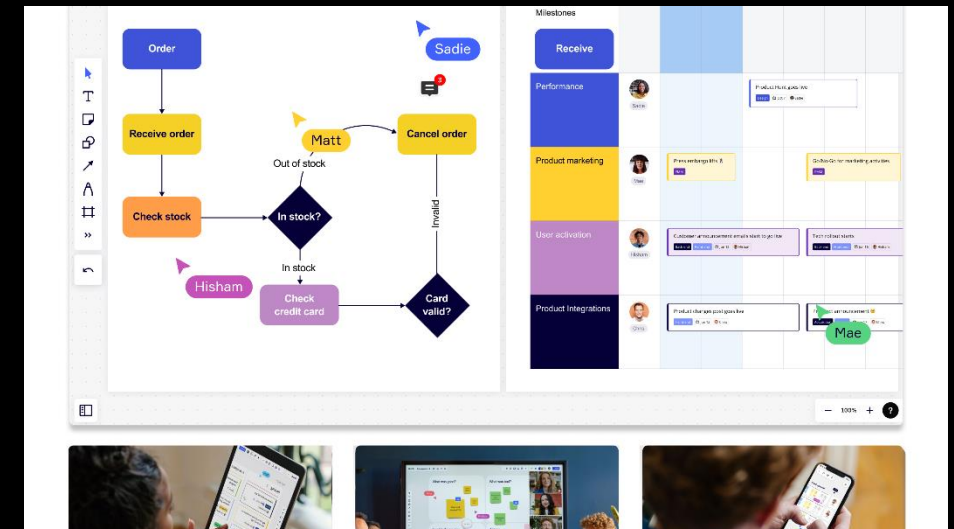
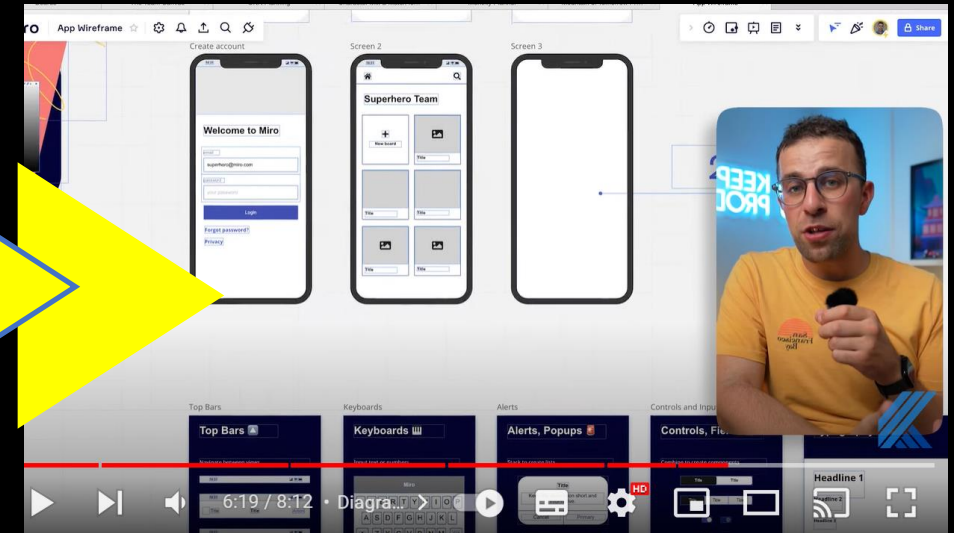
Pricing & Sign Up	Personal	Professional	Unlimited
	Free	\$9.95/MONTH*	\$19.95/MONTH*
Project schedules	1**	20	unlimited
Free Updates	✓	✓	✓
Email Support	✓	✓	✓
Collaborate Online:			

Miro.com

- Miro is the online collaborative whiteboard platform that allows teams to work together, from brainstorming with digital sticky notes to planning and managing **agile workflows using flowcharts, Gantt charts and wire-frame diagrams.**

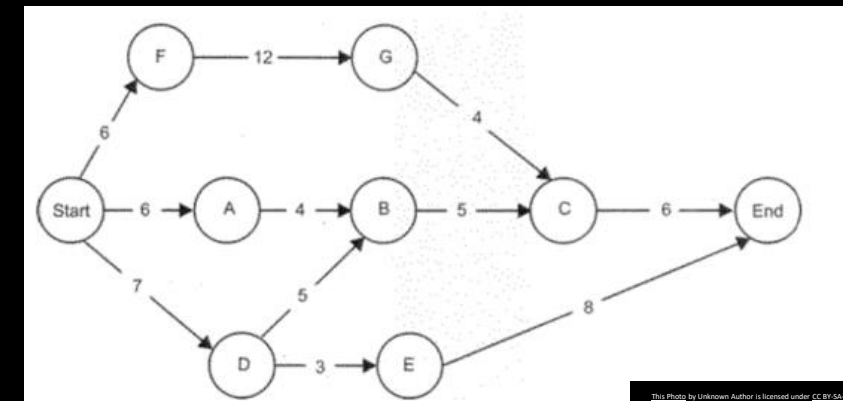
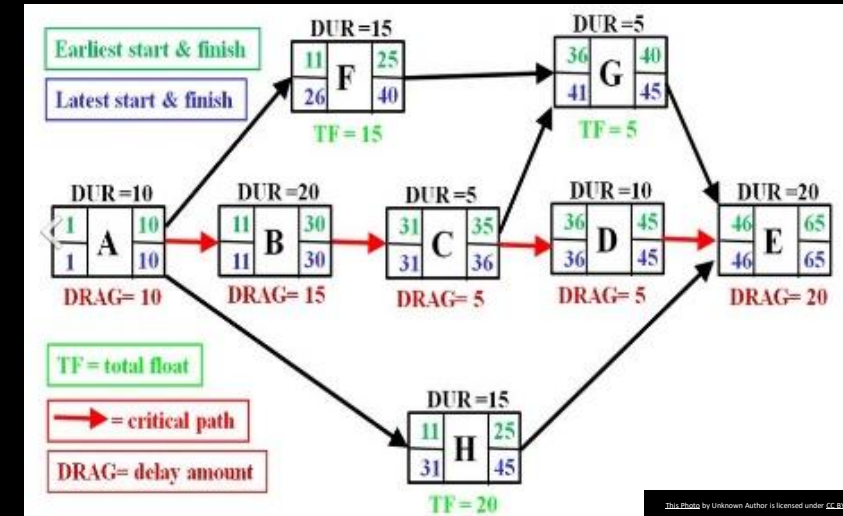
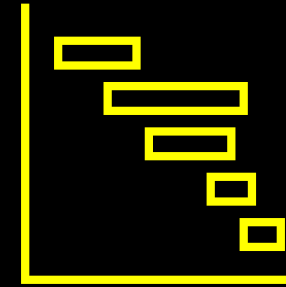
- **Bg. Play:** [Video Overview](#).
- Tip: You can Add a MS Plan in MS Teams as follows: [Video - How to create a plan with Planner in MS Teams](#).
 - Bg. MS Planner homepage: <https://tasks.office.com>.

Play

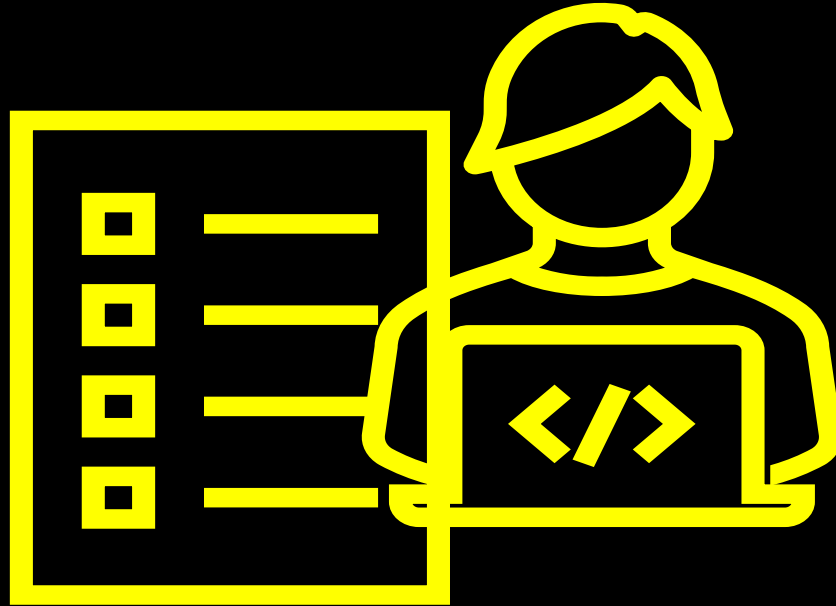


Project Chart choice

- **Chart:** Create a **Gantt chart** to plan your project timeline, stages and tasks. It is an easy, simple and straightforward project planning and management method appropriate for an academic project.
- **Chart types:** Although a Gantt chart is recommended, if you wanted to create a Project timeline / Roadmap, PERT (Program Evaluation Review Technique) or CPA (Critical Path Analysis) chart instead, you can.
- **Timeline:** The key focus is to create a '**Project Schedule and Timeline**' to list your Project stages, tasks to complete, and when to complete them.
 - **Selection:** It doesn't really matter which method you use, as long as it helps you to plan and manage your project. For small project's and due to their ease of use, recommended, and common use in industry for small projects, I recommend using a **Gantt chart**.



Project Development Methods

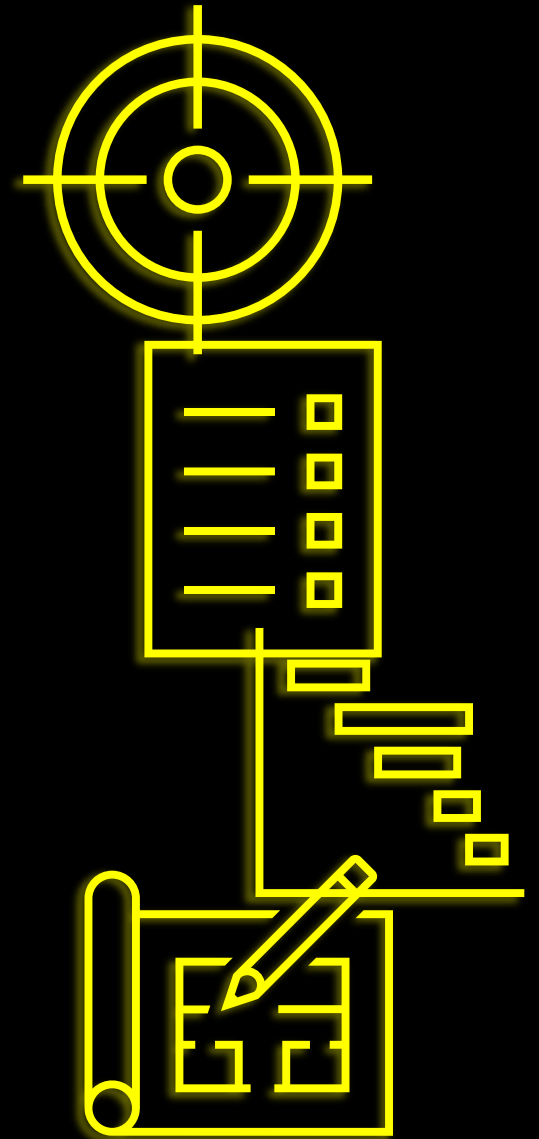


Development Methodology/Methods

There are many Development methods, such as:

- Waterfall Methodology
- The Software Development Lifecycle (SDLC)
- The Prototyping Process (TPP)
- Design Thinking (DTP)
- Rapid Application Development (RAD)
- FDD, Scrum, Kanban etc...

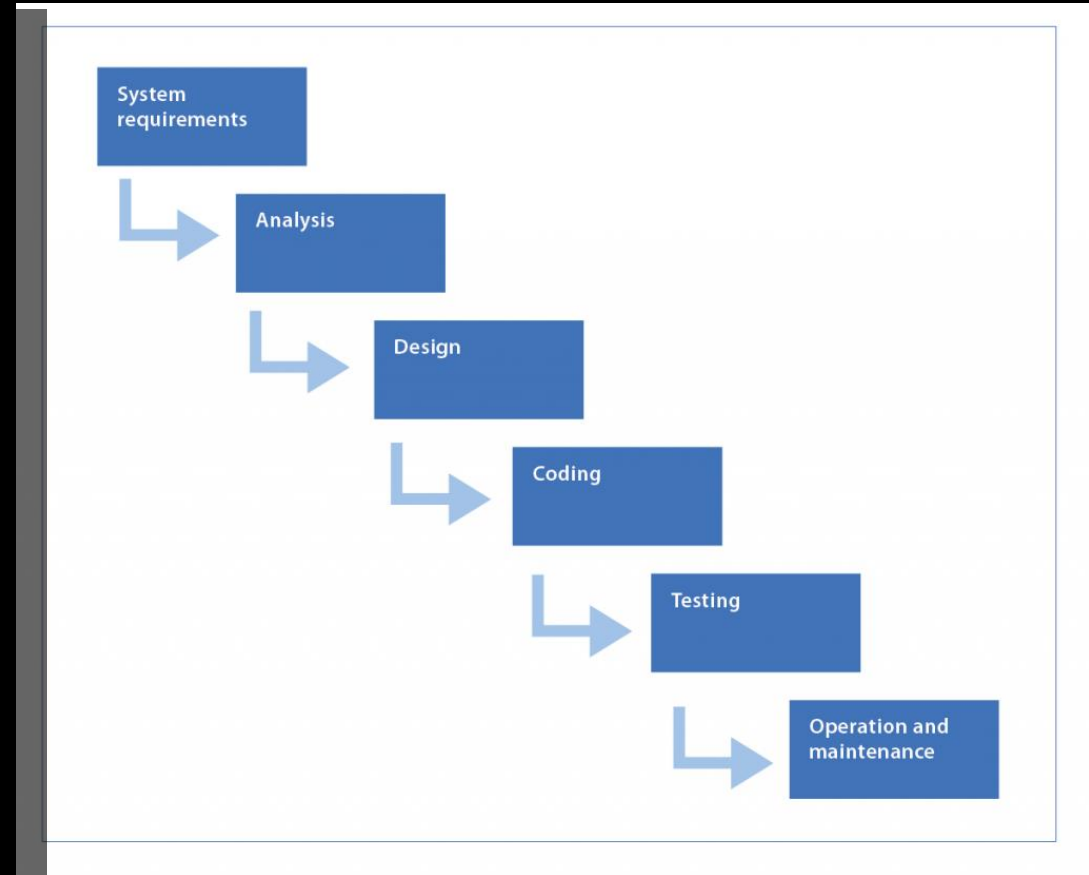
They all have very similar stages.



The Waterfall Methodology

- The Waterfall model / method / methodology is a sequential project development process.
- The Waterfall model is named as such because each stage of the model flows into the next.
- Flowing down like a waterfall through all phases of a project, which include: **Analysis, Design, Development, and Testing.**
- Requirements are gathered at the start, and the following project stages build on the other to achieve those requirements.

For more information on [the Waterfall Model please click here](#) and [here](#).

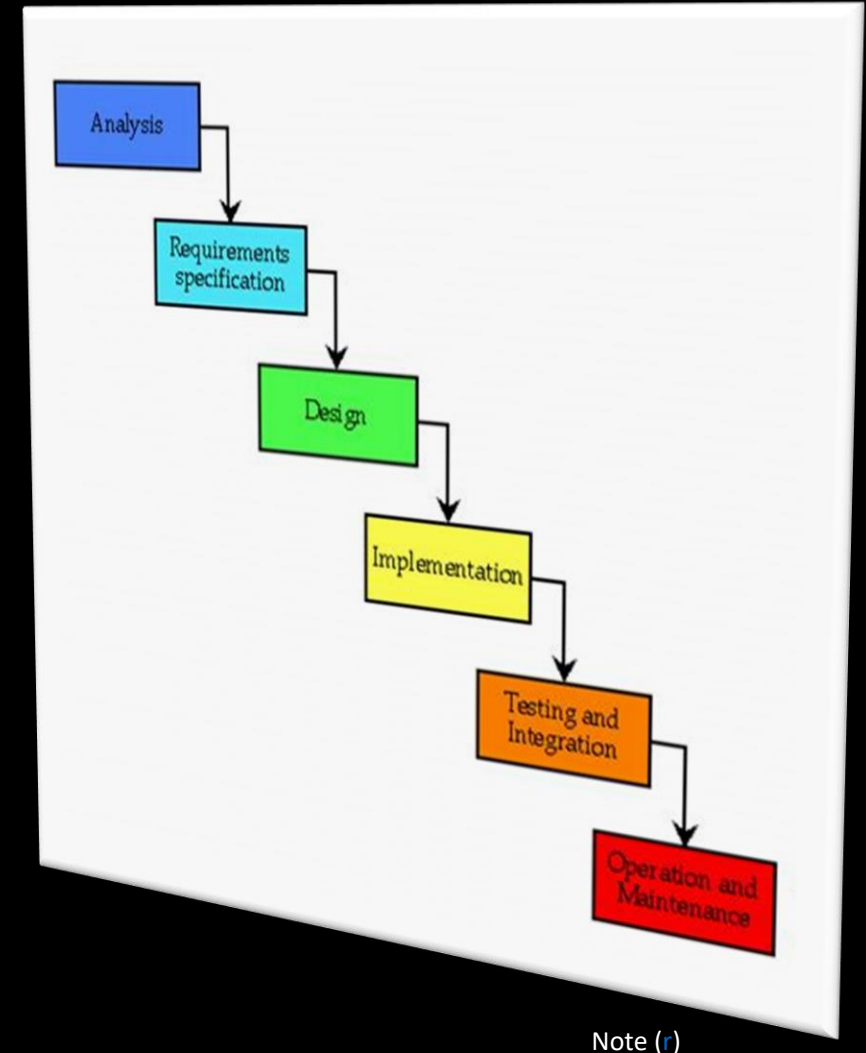


This Photo by Unknown Author is licensed under CC BY

The Waterfall Model

The original waterfall model followed phases in order:

- **System and software requirements:** capture product requirements in a document;
 1. **Analysis:** create requirements and models;
 2. **Design:** the architecture, schemas, flow charts, software patterns;
 3. **Implementation:** the development and integration of software;
 4. **Testing:** systematic testing and debugging;
 5. **Operations & maintenance:** the installation, support, and maintenance of a completed system.
- The waterfall model maintains that one should move to a phase only when its preceding phase is complete, reviewed and verified.
- Various modified waterfall models include variations on this process. These variations include returning to the previous cycle after flaws were found downstream, and returning to the design phase if downstream phases were deemed insufficient. So, there is flexibility here.

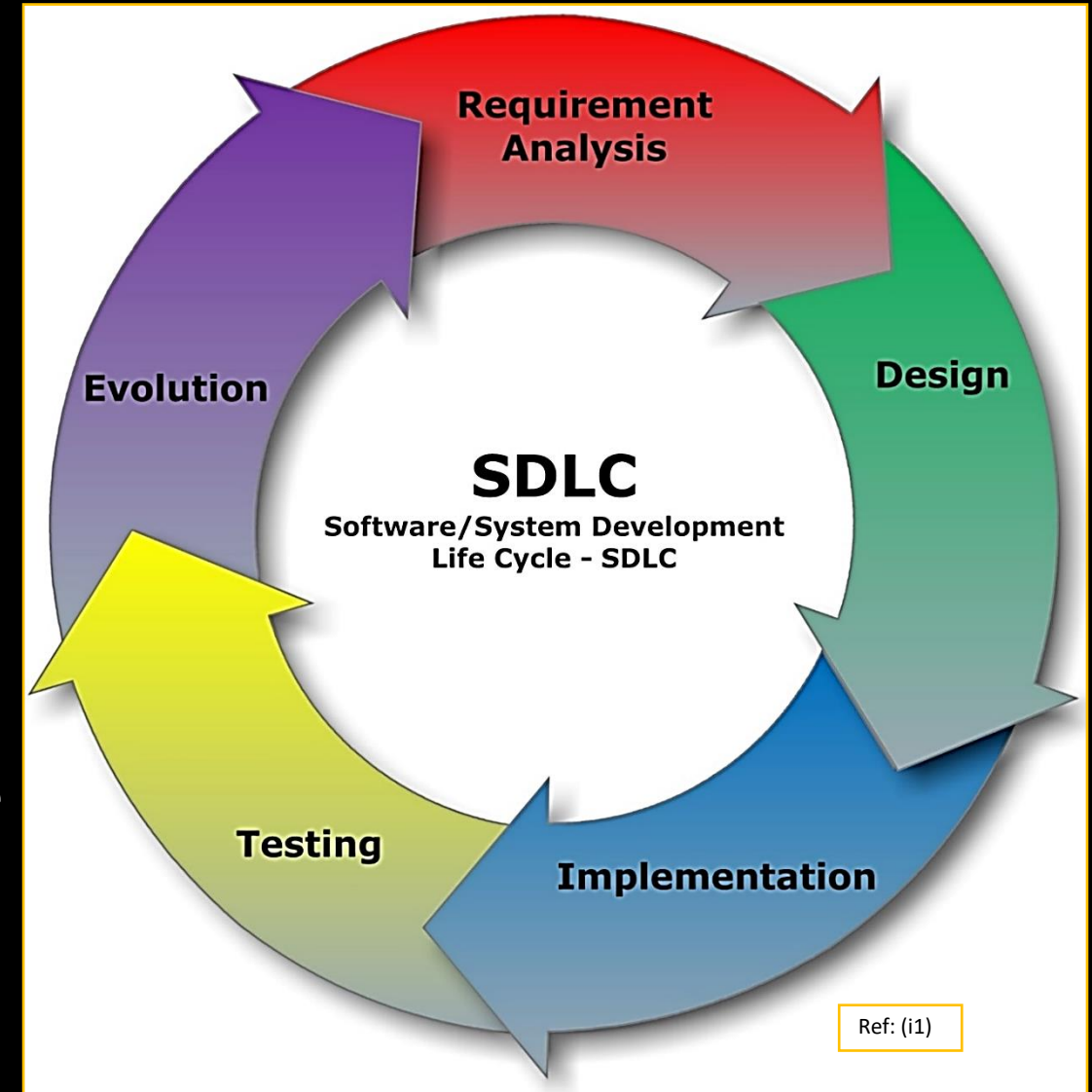


The Waterfall Model

- The Waterfall model was originally **perceived** as sequential not iterative so that when the (exhaustive) requirements were agreed they would not change (at least until the entire project was complete) post deployment.
- Extensive documentation throughout all phases.
- Popular in the 70s & 80s until the 90s when RAD & prototype methods emerged.
- Still used where emphasis is required on strict planning control and documentation.

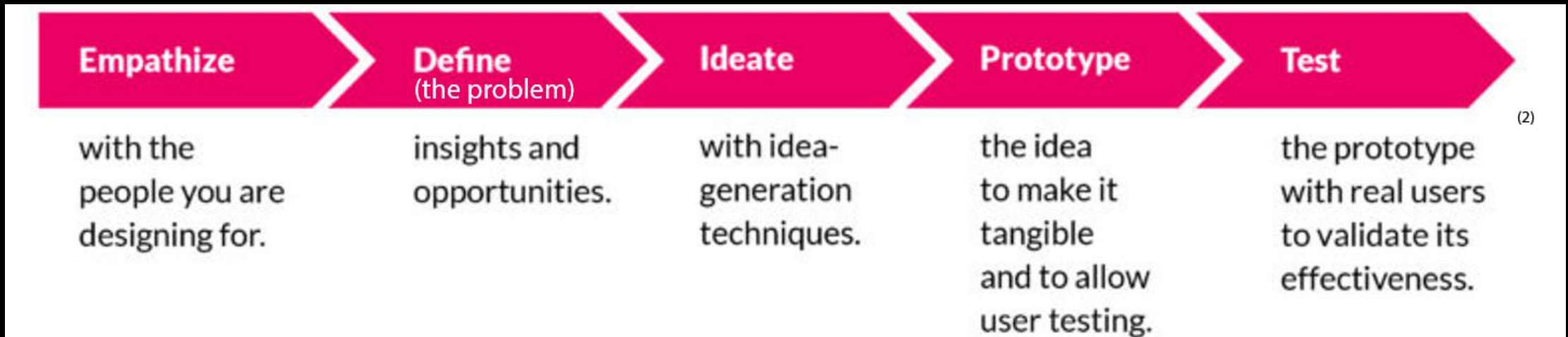
The Development Life Cycle

- The Systems aka **Software Development Life Cycle (SDLC)** is used to describe key stages in a project process.
- These stages include: **Project Analysis, Design, Implementation, Testing, and evaluation & evolution.**
- Traditionally used for Computer System/software development, it can be applied to all projects, and variations are used in many different industries.
- It is sequential but also flexible and iterative.



Design Thinking Process

- Similar to the SDLC Model, is the **Design Thinking Process**.
- **Design Thinking is all about finding a problem worth solving.** It is a systematic process to solving problems based on **empathetic** observation of people.
- It is a **process of Innovation**.
- Today, **Design Thinking** is popular in many industries including Design & IT/Computing sectors as a method of problem-solving and innovation.
- Similar to other project process models, it also generally includes 5 key stages.
- The DT process is **also flexible, iterative and loops back around**.
 - For more information about Design Thinking, it's History and origin please click [here](#).



Prototyping

- **Prototyping** is the activity of **creating incomplete prototypes** and **improved iterative versions** of what is being developed.
- **Prototypes allow users to test and evaluate early on in the project.**
- **Early feedback** at the Analysis & Design stage of the project improves the **prototype, final implementation** and user experience.
- Prototyping is a development method you could use for your project.

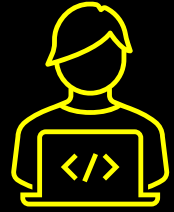
Prototyping

- Prototypes come in many forms, from a rough sketch on paper to an interactive simulation that looks like the final product.
- The **Rapid Prototyping Process** includes a three-step process, repeated as many times as needed:
 1. **Prototype:** Create a visual mock-up of your solution.
 2. **Review:** Share the prototype with users and evaluate to see if it meets their needs and expectations.
 3. **Refine:** Based on this feedback, identify areas that need to be improved or clarified. (r)
- Bg. For more information, see e-Handout: [A Beginner's Guide to Prototyping](#).

Software prototyping

- Software prototyping is the activity of creating prototypes of software applications i.e. incomplete versions of the software program being developed.
- Prototyping has several benefits:
 - The software designer and implementer can get valuable feedback from the users early in the project.
 - The purpose of a prototype is to allow users of the software to evaluate the product by actually trying it out, rather than having to interpret and evaluate the design based on descriptions.

The Prototyping process



The process of prototyping involves the following steps:

1. Identify requirements: Determine basic requirements including most important elements. Details can be left for later consideration and iterative refinement.

2. Implement the practical work and develop the initial prototype.

- Bg. There are different **prototyping methods** such as: Incremental, Throwaway, Xtreme or Evolutionary prototyping or just jump in and just get building. For more information please [click-here](#).

3. Test & Review: With relevant stakeholders (e.g. Dev Team and end-users) examine the prototype and provide feedback on potential additions/changes many times throughout the project.

4. Revise and enhance the prototype: Using feedback, the Requirements specification and prototype can be improved.

5. Repeat.

Note: This is a traditional 'Software prototyping / prototyping' process that is used in many different industry sectors.

Bg. Software Prototyping method

- The process of prototyping involves the following steps:
 1. **Identify basic requirements** - Determine basic requirements. Details, such as performance and security, can initially and typically be ignored.
 2. **Develop initial prototype** - The initial prototype is developed that typically includes only user interfaces with bare functionality.
 3. **Review** - Customers / end-users, examine the prototype and provide feedback on potential additions or changes.
 4. **Revise and enhance the prototype** - Using the feedback the specifications and the prototype can be improved.
 - Bg. Negotiation about what is within the scope of the contract/product may be necessary
 5. **Iterate**: If changes are introduced then a repeat of steps #3 and #4 are undertaken. (t)

Contemporary Tech Development methods

Generally New Software development & IT Management methods value: e.g.

- Flexibility Change & Innovation
- Continuous improvement
- User / Customer focused and involvement
 - People over Processes.
- Open holistic collaboration with all stakeholders (e.g. Business owner / client)
- Empowerment: Bottom-up employee involvement
- Business & IT collaboration
- Small milestones
- Practical testing over planning and documentation
- Sustainable balanced working.

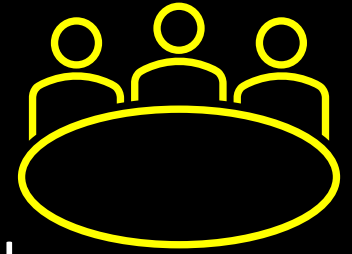


Agile Development

- Up-to-date software development methodologies focus more on people / users, embracing change and flexibility than on strict planning methods and tools.
- Agile dev methods reflect the new values of focusing on people and embracing change.

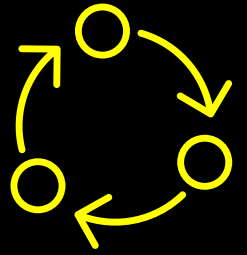


Agile Development



- Agile development methods include **requirements gathering** and prototype/feature improvement through cross-functional team collaboration and collaboration with their users/client/customers.
- Key aspects of agile include: Early implementation/delivery, frequent meetings, flexible/adaptive planning, evolutionary development & continual improvement and flexible changes in requirements.
- Agile methods focus on iterative quick start prototype development, frequent user feedback and iterative (cyclical) feature/prototype/product improvement.
- The principles of Agile can be used by individuals on individual projects. Typically however, Agile development methods are commonly used by Software Development **Teams** to build software collaboratively.

Rapid Application Development



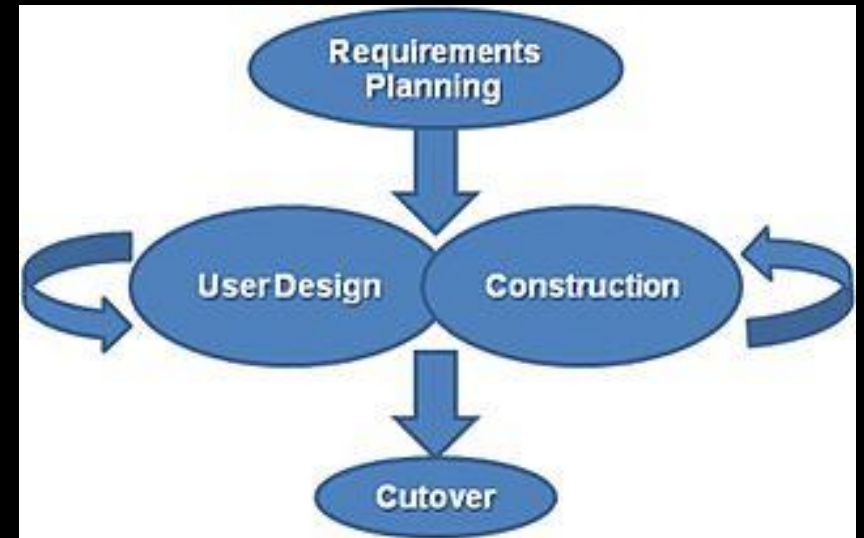
- According to many sources, the Waterfall method of software development had its roots in conventional engineering. And that it involves very carefully planned and absolute sequential stages that had to be progressed through in a linear fashion and did not have the flexibility of returning to and updating earlier stages.
- Nevertheless, to address this, in the early 1990s, the early creators of RAD saw that software engineering was different to traditional engineering in that software wasn't a fixed structure such as a bridge or building. Software was soft not hard, it was malleable/flexible and could be changed/updated to suit the needs of the user.
- And that software development was not limited to traditional methods of engineering. In that less emphasis could be put on initial and absolute planning and absolute progression between project stages, and that software engineering is more of an adaptive evolving process that could be iterative.
- Rapid Application Development or RAD "focuses less on rigid plans and more on fast, iterative cycles of prototyping ... [and] relies on user feedback throughout."

Bg. Definition: Rapid Application Development

- RAD is an agile adaptive software development model, that is based on prototyping and quick feedback with less emphasis on planning. With rapid application development, developers can change the design, add functionality, and keep reiterating without having to start from scratch each time. (r)
- Rapid Application Development is an agile framework focused primarily on rapid prototyping of a software product, frequently iterating based on feedback, and continuously releasing updated versions of the software product to the market. (r)

Rapid Application Development

- The initial phases of RAD included: 1: Requirements planning. 2: User design. 3: Rapid construction and 4. Cutover (Deployment)
- The similar but updated RAD model includes 4 key steps:
 1. Requirements definition
 2. Initial Prototype
 3. Iterative Development (with user feedback)
 4. Deployment



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

RAD: Rapid Application Development

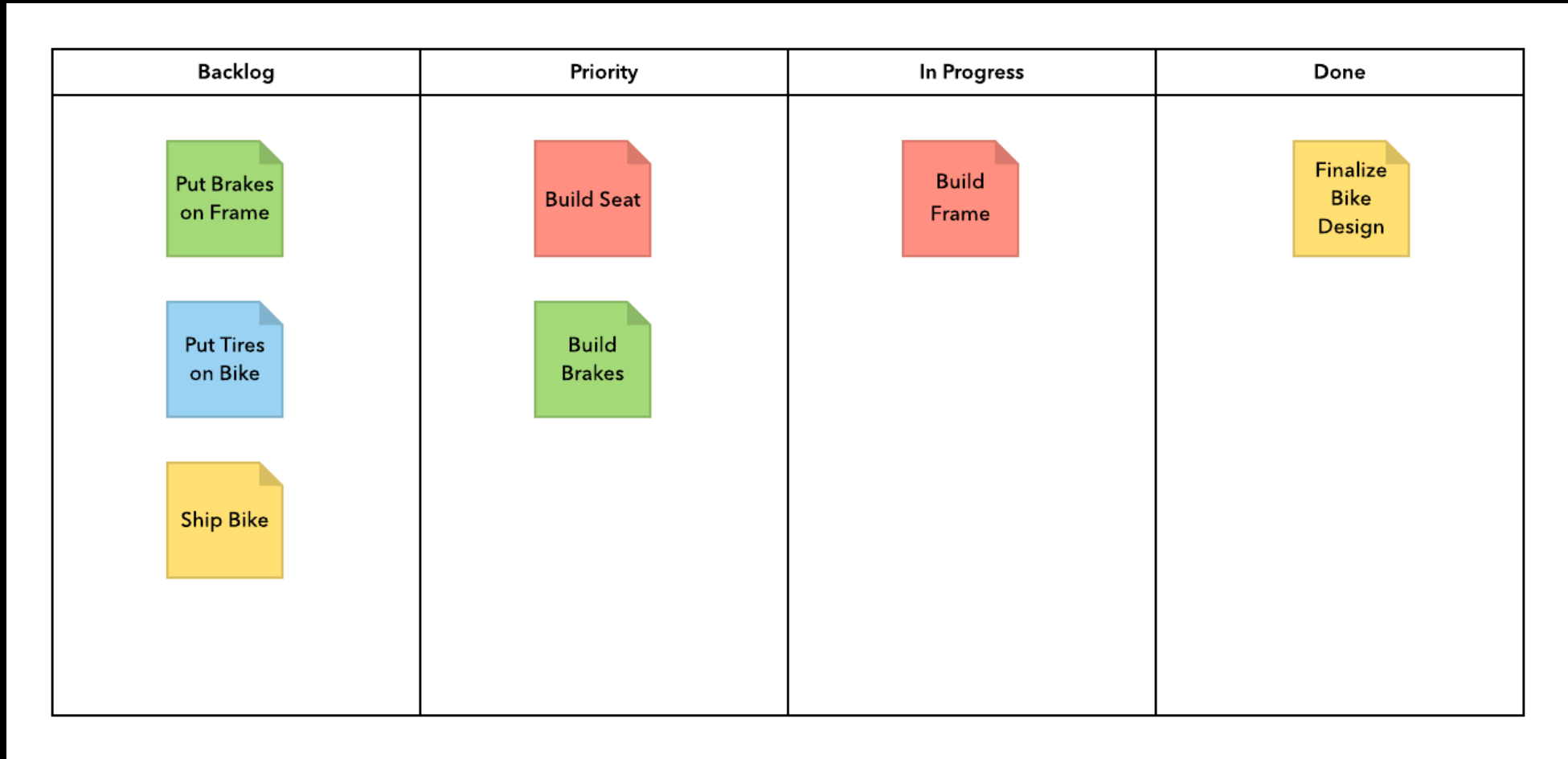
- RAD is user-focused, iterative and emphasis practical prototype over process.
- However documentation was typically still extensive at the design phase e.g. Vast array of UML diagrams for design / systems analysis.
- Popular today for creating managing and maintain large complex systems.
- Modified versions of RAD with less focus on documentation are popular today for creating smaller scale software projects.



Agile: Feature Driven Development (FDD), Kanban and SCRUM

- Some other example Agile development methods include FDD, Kanban, SCRUM.
- **Feature-driven development (FDD)** is an agile software development framework, that focuses on identifying required 'features', assigning features to personnel/teams and building software, feature by feature.
 - Bg. For an overview of the [FDD framework please click here](#).
- **Kanban** (developed by Japanese company Toyota) is a project scheduling & development method that uses a **visual 'Kanban board'** that moves identified required product features known as a 'backlog list' through typical project stages to production/deployment.
 - Bg. For a Kanban [overview](#) and example [tools](#), please click the preceding links.
- Similarly, **SCRUM** is a team-orientated requirement/feature focused agile software development method for organising and managing a software development project.
 - Bg. For an overview of the [SCRUM framework please click-here](#), and for a [Video of SCRUM click here](#).

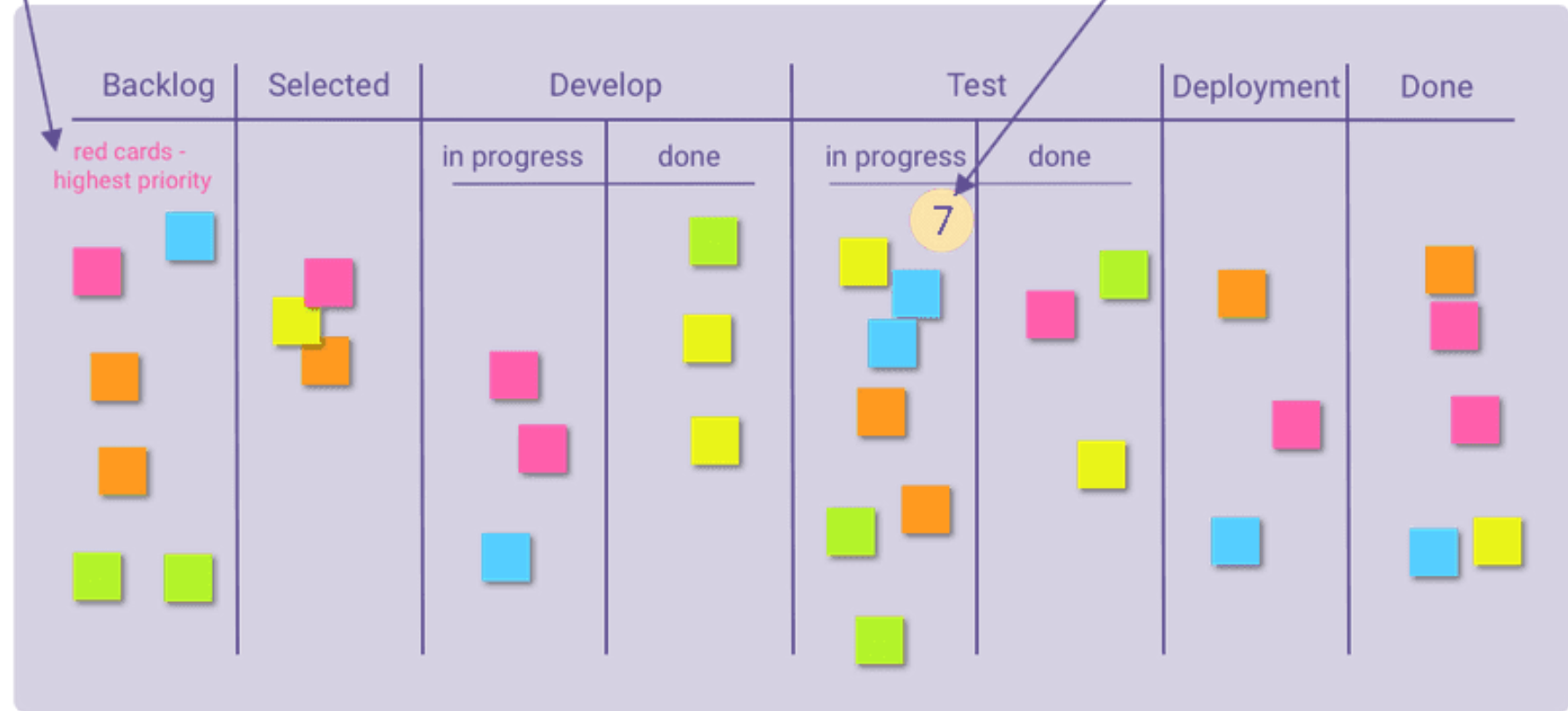
Example: Kanban board



Bg. Kanban board

Explicit guidelines

WIP limit

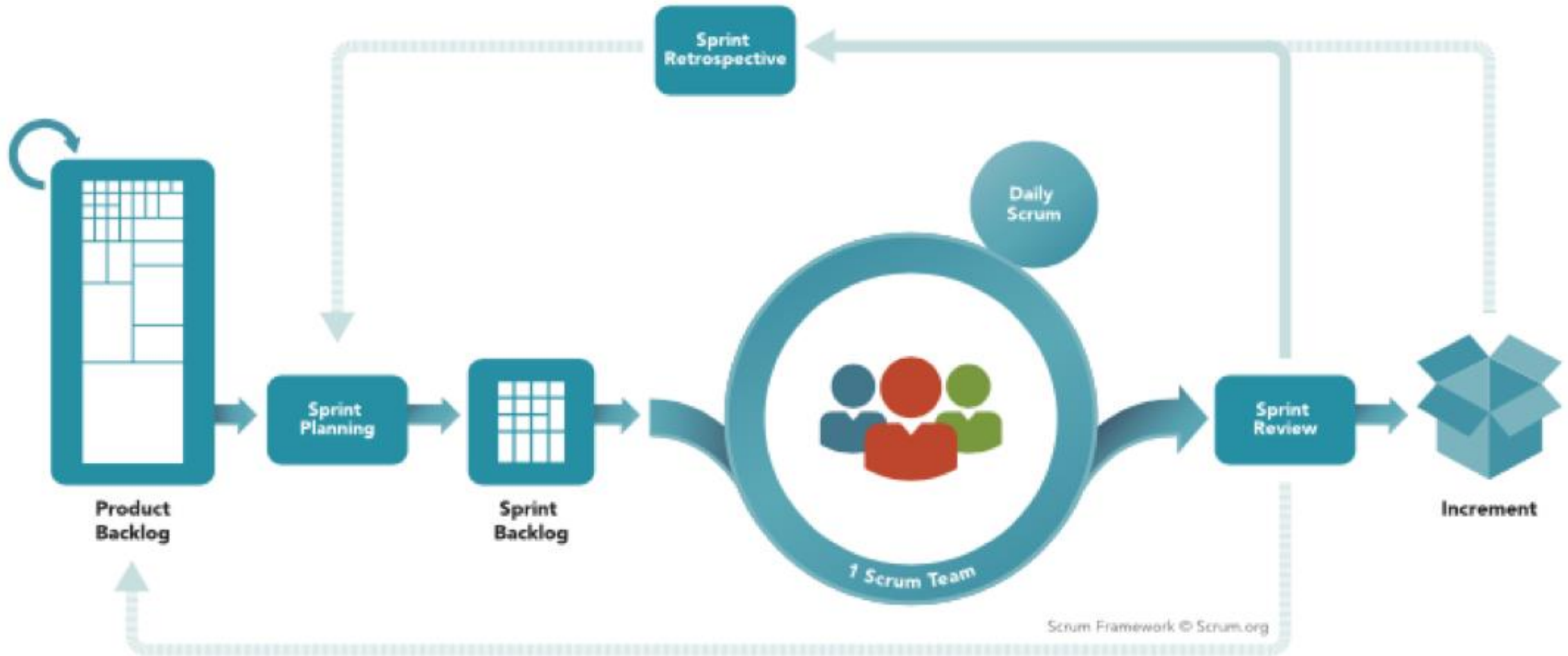


Bg. FDD & Kanban

New	Analysis		Design		Development		QA	
	In Process	Done	In Process	Done	In Process	Done	In Process	Done
Feature	Feature	Feature	Feature	Feature	Feature	Feature	Feature	
Feature	Feature	Feature	Feature	Feature	Feature	Feature	Feature	
Feature	Feature	Feature	Feature	Feature	Feature	Feature	Feature	
	Feature		Feature		Feature	Feature	Feature	
					Feature	Feature		

SCRUM FRAMEWORK

The **Daily Scrum** is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. (r)



Ref: <https://www.scrum.org/resources/what-is-scrum>

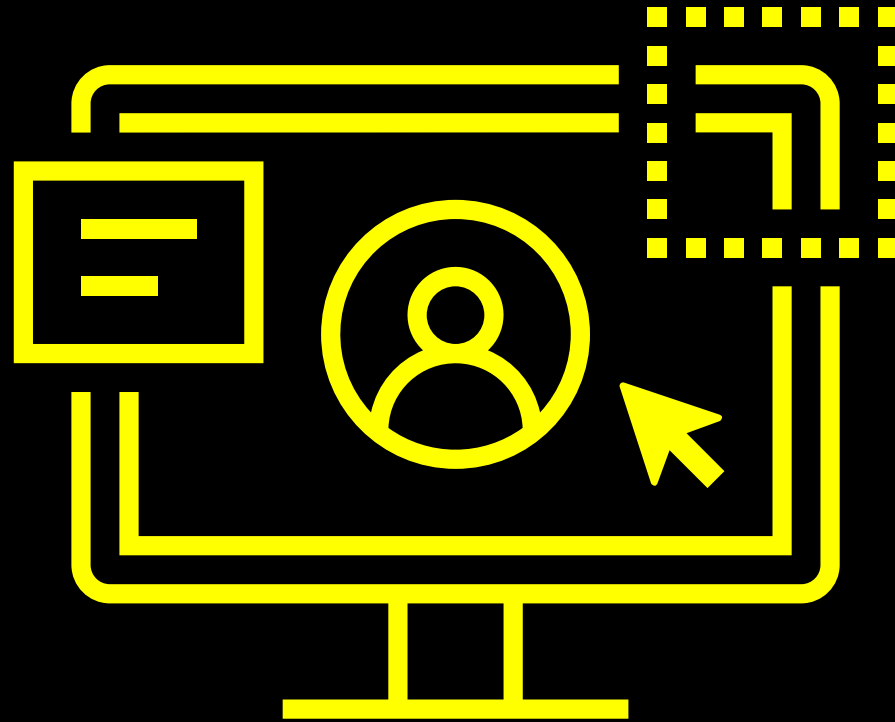
Further information: <https://www.scrum.org/resources/what-is-scrum>

Project Methods & the Assignment

- Initially, for the purpose of a **Computing Academic Research proposal**, all that is needed is a basic understanding of the **typical/traditional project process phases** including:
 - **Requirements Analysis > Design > Development > Testing and Evaluation**
- You can choose any Development method (e.g. SDLC, Prototyping, RAD, FDD) you feel is appropriate for your project.
- You can select a Development method and modify it to suit your project. Or select and combine aspects of 2/3 methods such as SDLC and a Kanban board. This is common practice in industry.
- Whether you select 1 as it is, customise 1 or select 2 and modify them. Just include a little description to justify your selection/customisation. That's ok.

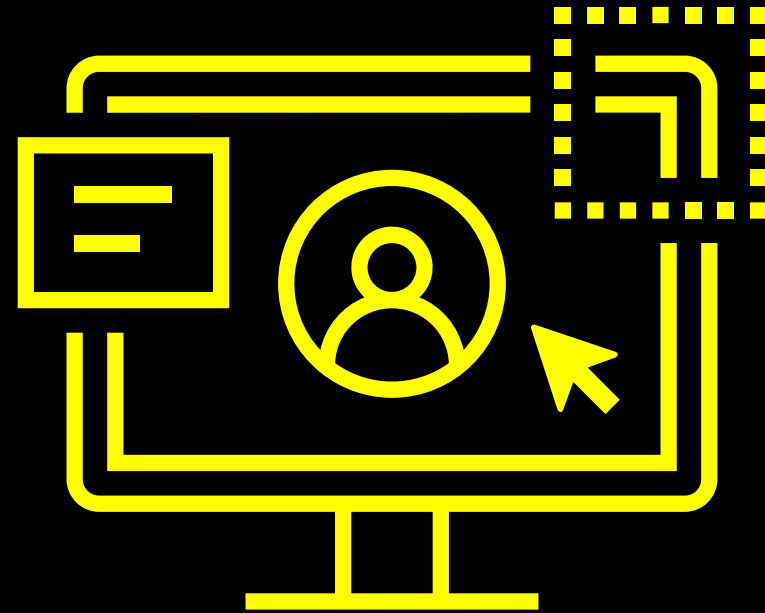
Bg. Agile development methods are covered in more detail on some programmes e.g. on the Software engineering programme, there is a module where students are grouped into teams to build software using an agile process.

Project Design



Project Design

- The following slides introduce some key terms and concepts commonly used in the design stage of a Computing project.

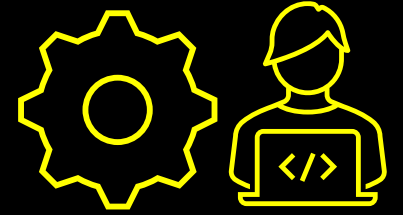


Design tips: Overview

- The upcoming slides, listed below, contain Design options that may be useful for the Design section of your assignment and project. You will be doing some of these in other modules. A brief overview is provided here.
- You do not need to include all of the following, due to the different various types of projects, different design methods are covered for consideration, use 2/3 methods may be suitable and can be used for your assignment/project.
- Aim to include approx. 2 (3 if needed) types of design work. The following slides will cover:

- | | |
|---------------------------------|---------------------------------------|
| • 3-Tier Design considerations. | • Interface & prototype design tools. |
| • Requirements Specification. | • UI Design, Wireframes & Mock-ups. |
| • Full Stack Design Model | • Network Mapping tools. |
| • Minimum Viable Product (MVP). | • Product design specifications. |
| • Prototyping on Paper (PoP). | • Design work. |

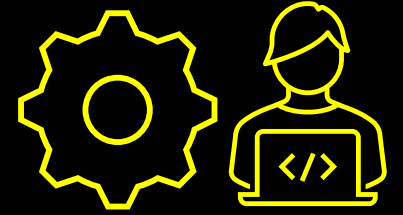
Requirement Specifications & Prioritisation



We will consider two types of Specifications: Requirements & Product specifications:

1. **Requirements / Feature specifications:** Are suitable for the creation of new Computing prototypes.
2. **Product specifications:** Are suitable for the creation of new hardware products e.g. a component list.

Requirement Specifications & Prioritisation



- **Requirement specifications** can be used for hardware or software prototypes and should be **prioritised**. This will help you to focus on what is most important.
- When prioritising your requirements, consider the most important features of your project's implementation aka **MVP Minimum Viable Product**.
- Many methods can be used to **prioritise** your list e.g. By number (1, 2, 3, etc..), High, Med. or Low, or the MoSCoW* method.
 - * M for Must haves, S for Should haves, C for Could haves and W for Won't haves.

MVP - Minimum Viable Product



- **MVP Minimum Viable Product** is a prototype/product with **just enough features to satisfy early customers**, and to provide feedback for future/iterative development.
- What is your MVP? i.e. the most important features for your project implementation.
- Consider your MVP to help you identify your top-level/high priority requirements.
- MVP is a **Lean Start-Up method**. Termed by early Tech Entrepreneurs based in Silicon Valley California and adopted by the most successful Tech companies such as Google e.g. Gmail launch and all other Google Apps.
- It is now used by Tech Start-Ups across the world.

Requirement Specification template



- A Requirements specification is a description of the **features** for a proposed prototype to be developed. It outlines the **SCOPE** of the prototype.
- A Requirement often **starts with a unique Number, a Name, followed by a priority status** and **brief description** that outlines the **purpose** of the requirement.
- A Requirements specification outlines **Functional (Fr)** and **Non-functional** requirements (NFr).

Unique number	Requirement Name
Priority status P:1	Requirement brief description.

Requirement Specifications



- You can have a separate requirement list for **Functional** and **Non-functional** requirements.
- A functional requirement outlines something the system should do/perform.
- Non-functional requirements outline all other non-functional aesthetic/operations/security activities that are required by the system.
- You may have **different types of Non-functional requirements** and they can be grouped together or partitioned into e.g. Design requirements, Performance req., Security req. etc... for the assignment you can have one simple list, that lists key features.

Fr:1	Requirement Name
P:1	Requirement brief description.

A Simple Requirement list (Keep it Simple)

Fr:1	Requirement Name
Mh	Requirement brief description.
Fr:2	Requirement Name
Mh	Requirement brief description.
Fr:3	Requirement Name
Mh	Requirement brief description.
Fr:4	Requirement Name
Mh	Requirement brief description.
Fr:5	Requirement Name
Ch	Requirement brief description.
Fr:6	Requirement Name
Wh	Requirement brief description.

Bg. Basic Requirement example for a small project

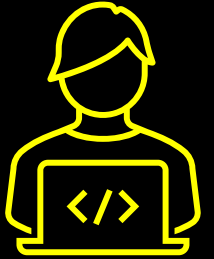
Requirement ID	Requirement Statement	Must/Want	Comments
FR001	The website will have a homepage which greets new users and introduces them to the different features the website offers.	Must	
FR002	The website will have a sign-up and log in function to allow users access to unique pages separate from visitors.	Must	
FR003	Profile page will be added for users who have logged in.	Must	
FR004	Exercise page that lays out a routine for users to follow.	Must	This can be found here - https://www.reddit.com/r/bodyweightfitness/wiki/kb/recommended_routine
FR005	Help page will be available for users to find answers to questions that are likely to be asked.	Must	These will be found from the reddit page that the exercises are sourced from.
FR006	A list of alternative exercises that will help users explore different types of fitness hobbies.	Must	These can include running, hiking, yoga and swimming.
FR007	Food page that informs users on both weight loss and bulking.	Must	Clearly include a disclaimer to state that the advice is not from a medical authority.
FR008	Dynamic features for each element testing a range of screen sizes to make sure the content is appropriate for mobile, laptop and desktop users.	Must	
FR009	Gamification feature rewarding users for progression.	Must	
FR010	Easier routines for users who want to do less exercise.	Want	
FR011	Calendar for users to schedule workouts	Want	
FR012	Notification feature to remind users to workout or to take a rest depending on time and day.	Want	

Bg. Further reading: Requirement Specifications

For further information and examples, please click below:

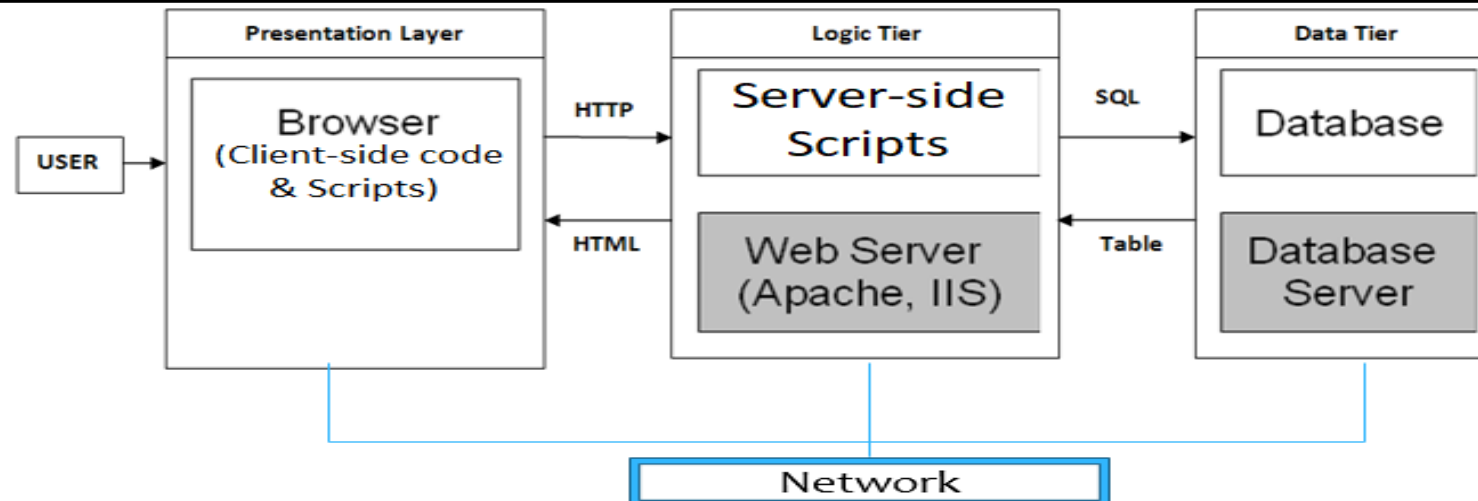
- [What is a Functional Requirement in Software Engineering? Specification, Types, Examples \(guru99.com\)](#)
- [Functional Requirements \(Functional Requirement Specifications, Functional Specs, FRS, FS\) \(ofnisystems.com\)](#)
- [A Guide to Functional Requirements \(with Examples\) \(nuclino.com\)](#)

Multi-tier / Full Stack Design Model

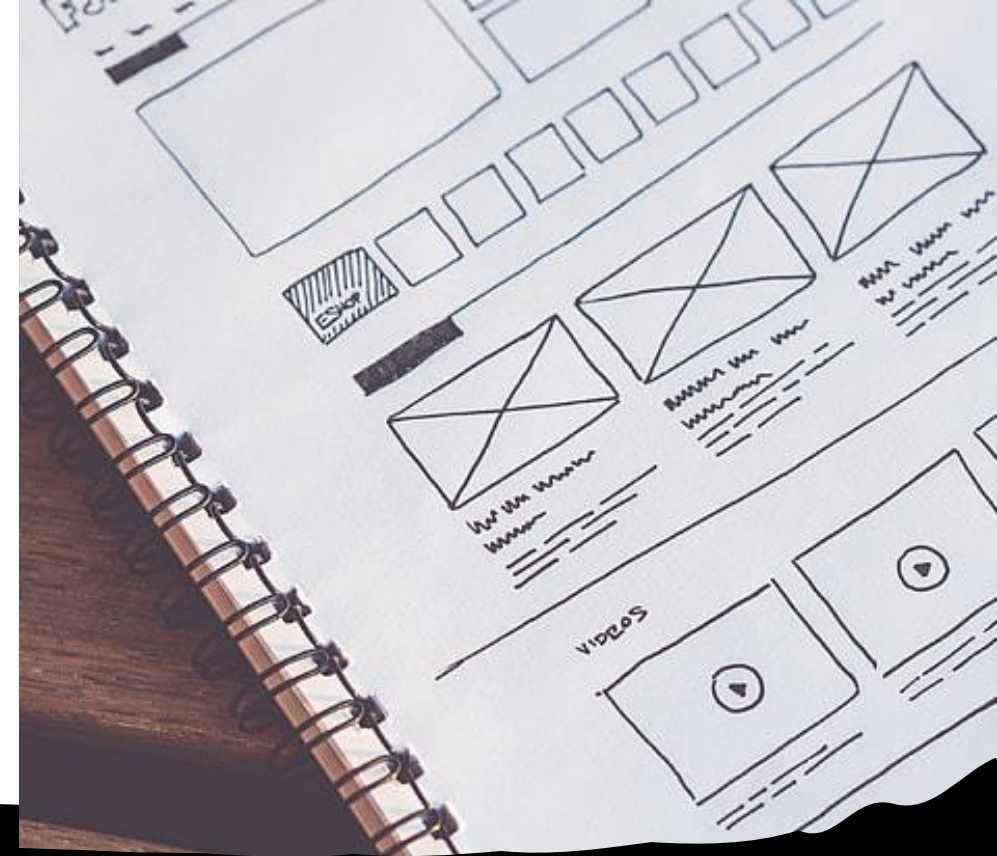
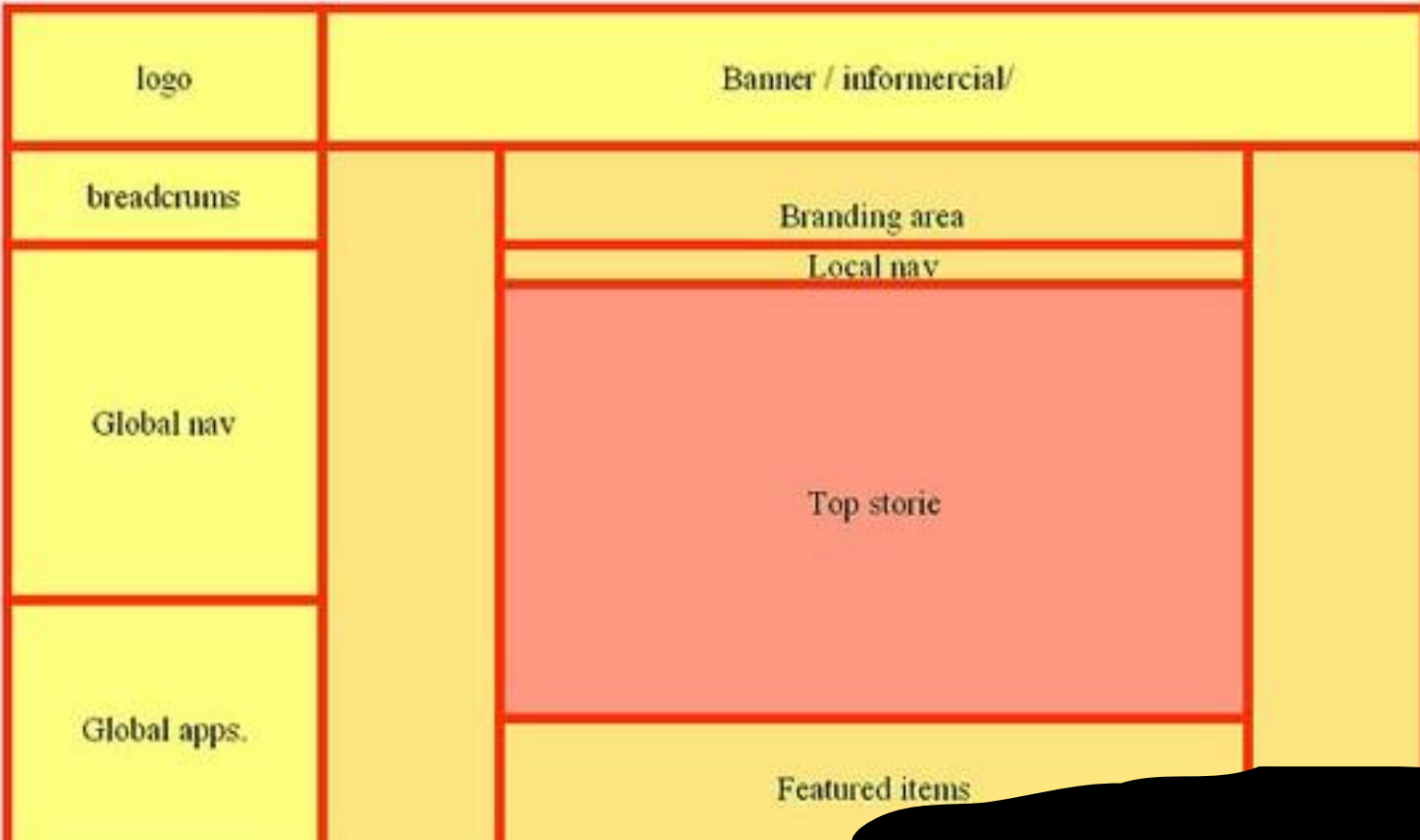


The table below lists some traditional design methods that you could use.

<u>Presentation tier</u>	<u>Logic/Server/Dev. tier</u>	<u>Data tier</u>
• Wireframe designs • Storyboard diagram • Colour & font scheme • UI Mockups • Branding (Logos etc)	• Pseudo code • Flow chart • UML diagrams • Identify needed Server-side/dev tools	• Database Schema (Normalisation & E-R Entity Relationship Models. • Identify needed Db Software.
• Hardware System/Architecture specification. * Network Map. • Server & Network Security requirements.		

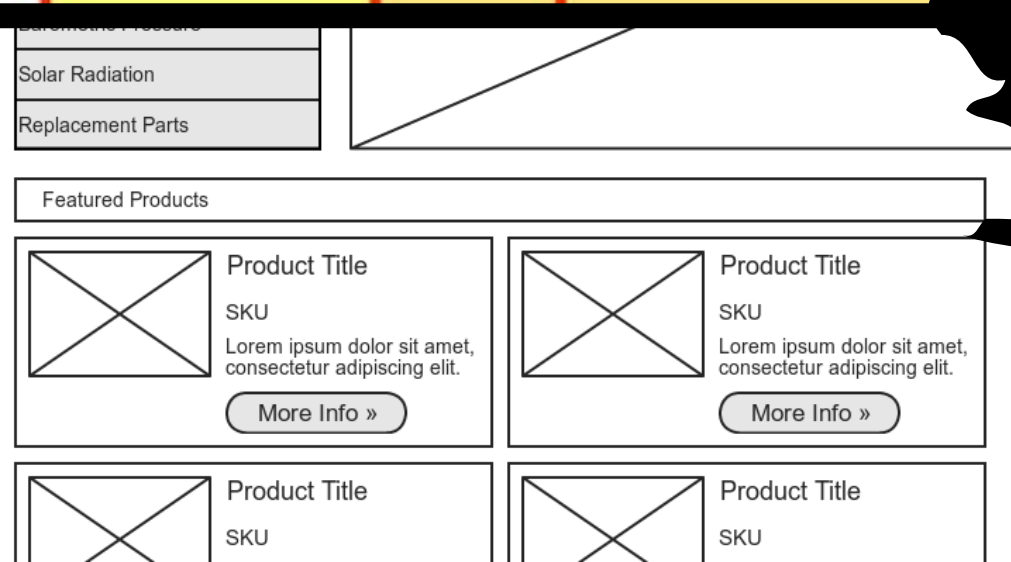


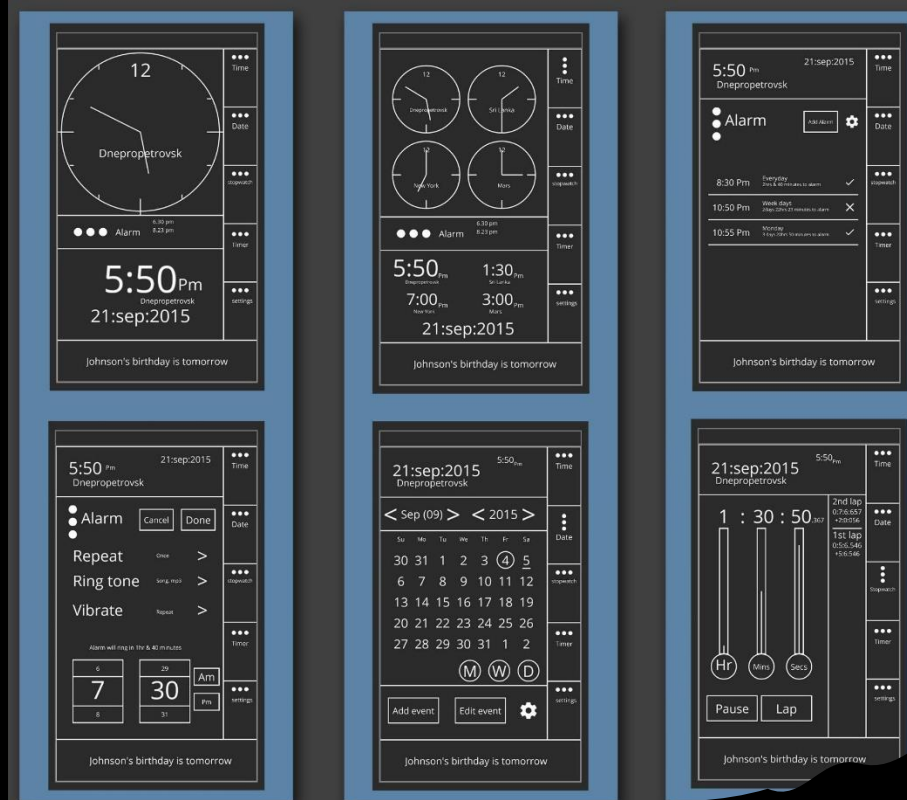
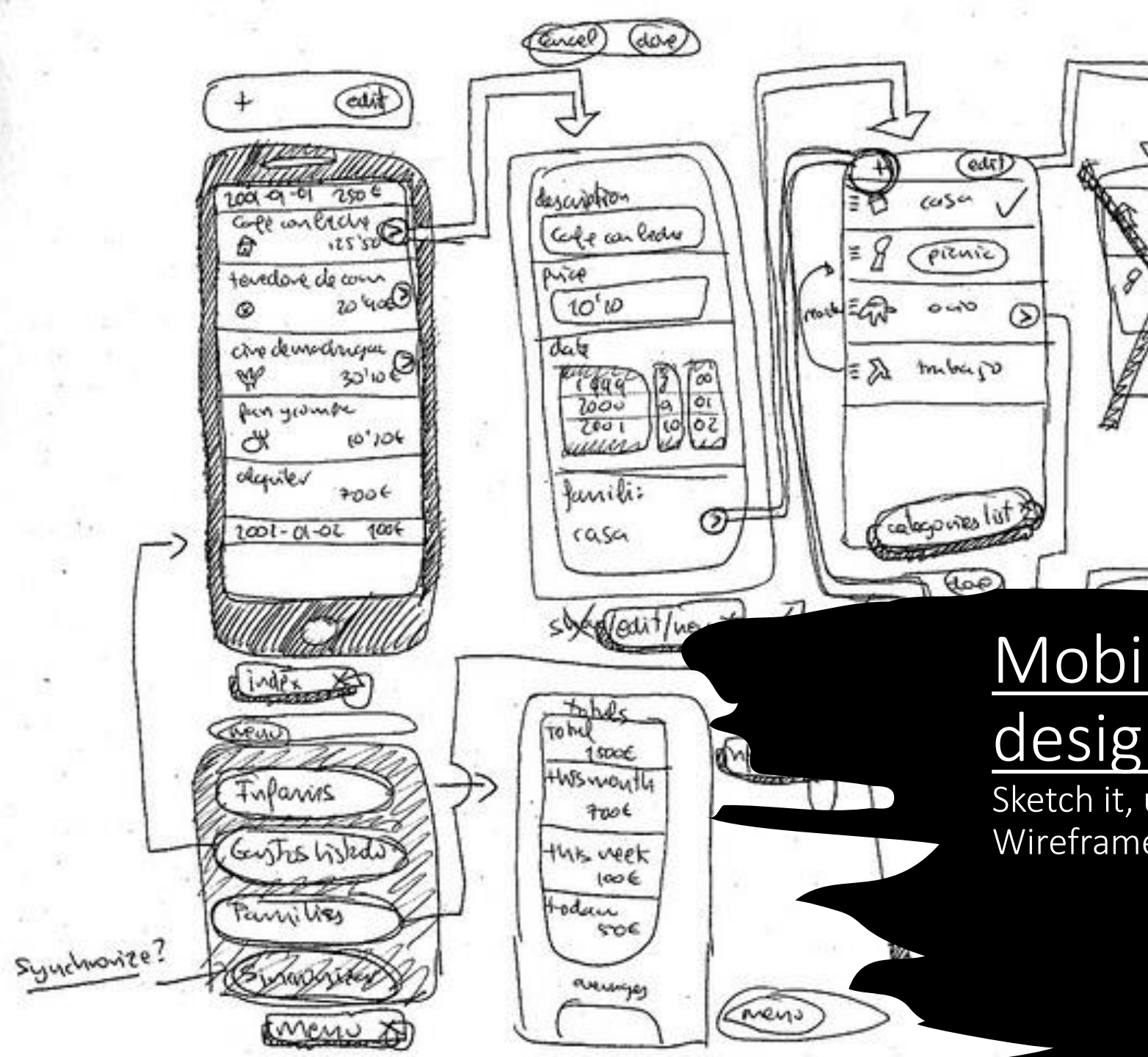
* Mock-ups are a proposed model or structure, for experimental design purposes.



Wireframe designs

Sketch it, use Word, Paint, a Graphics package or UI Wireframe design tool.





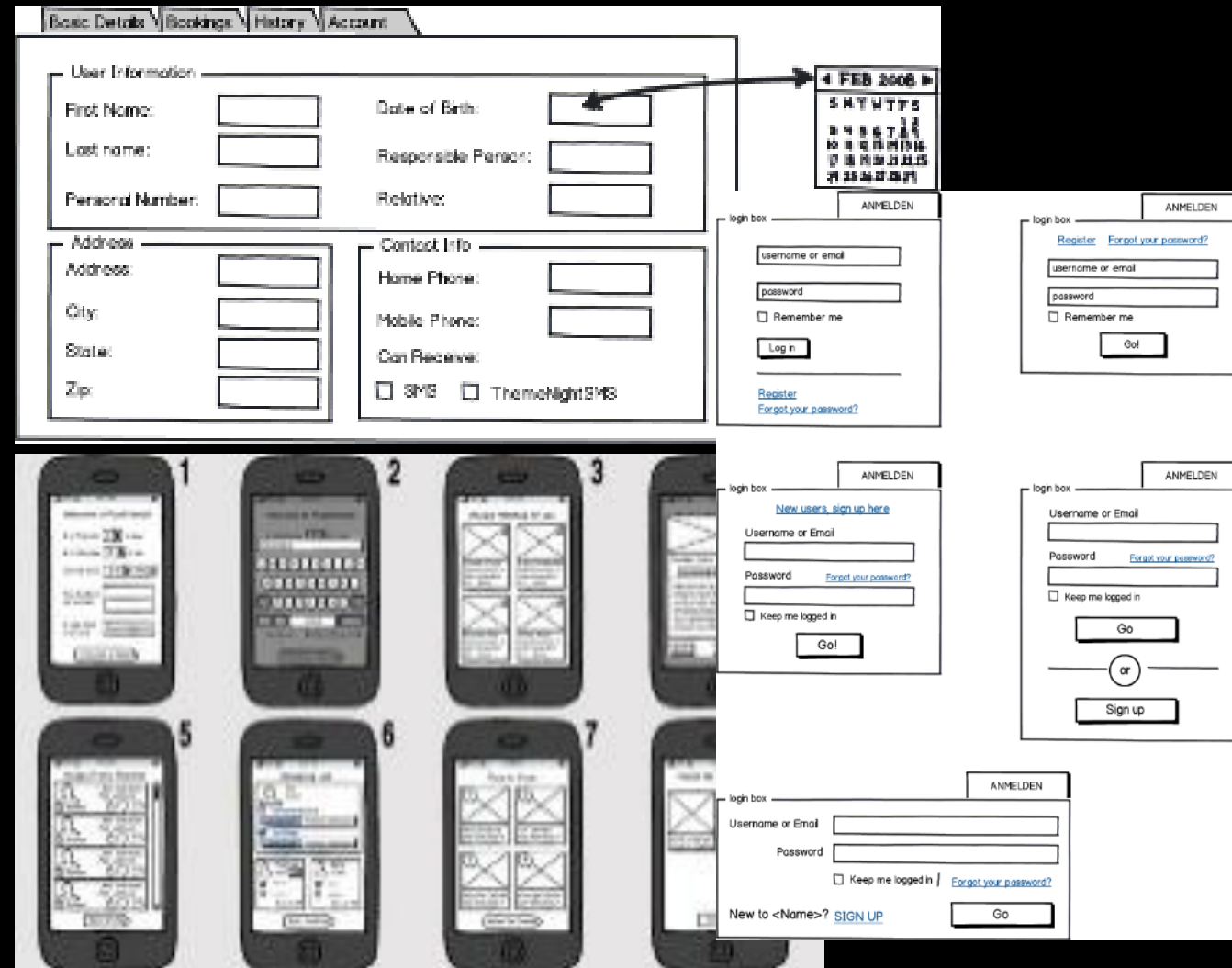
Mobile App. Wireframe designs

Sketch it, use Word, Paint, a Graphics package or UI Wireframe design tool.

Example Design tool: Balsamiq Wireframes



- **Balsamiq Cloud: User Interface (UI) Design Wireframe tool.**
- If you are building an App. you may find **Balsamiq** useful at the Design stage. For further information:
 - 1min. Video intro: [click-here.](#)
 - Website: <https://balsamiq.com/>.
 - Cloud and desktop versions: <https://balsamiq.com/wireframes/>
 - Learn to Wireframe: <https://balsamiq.com/learn/>.
- + You can put it on fish and chips ;-).



Bg. Balsamiq alternatives:

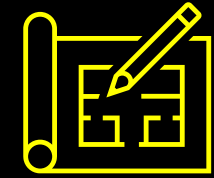
Balsamiq no longer has a free level, it is available as a 30 day trial, on campus (License via UI lecturer), there are many free alternatives, such as [Pencil Project](#), [Wondershare Mockitt](#), [Zeplin.io](#), [Figma.com](#), [PenPot.app](#), [Lucidchart](#), [Framer.com](#), [Moqups.com](#), [Uizard.io](#) and more.

Review these options and test 1/2 that are suitable for your project.

- [**Uizard: Great tool for wireframing, Mockups & prototyping.** *****](#)
 - [+ FREE FOR STUDENTS.](#)
- [Moqups - Online Mockup, Wireframe & UI Prototyping Tool](#)
- [Lucidchart - Intelligent Diagramming](#)
 - [5 Best Balsamiq Alternative for Free](#)
 - [Free Balsamiq Alternatives: Free Wireframing and Prototyping Tools](#)

Bg. Note: Not free but a good commercial Design tool for Web/Mobile Apps:
[Mockplus - Design, Prototype & Collaborate better and faster.](#)

Prototype Design: [Uizard.io](https://uizard.io)



[Uizard | App, Web, & UI Design Made Easy | Powered By AI](https://uizard.io)

- Uizard is a UI design tool for creating wireframes, mockups, and prototypes.
- Uizard also allow you to convert hand-drawn sketches into wireframes. And Transforms wireframes into prototypes.
- Uizard has many pre-made design templates and elements that you can drag, drop.
- Uizard also allows online Team collaboration.
- And has AI features .g. to screenshot a design you like and it will create a customizable version for you to edit for your own App.
- Uizard is releasing an AI text to prototype design feature called Autodesigner ([Show: video](#)). Type text & it will create a design for you.
- Uizard has a free, Pro and business [pricing structure](#).
- Key link: [Uizard.io](https://uizard.io)

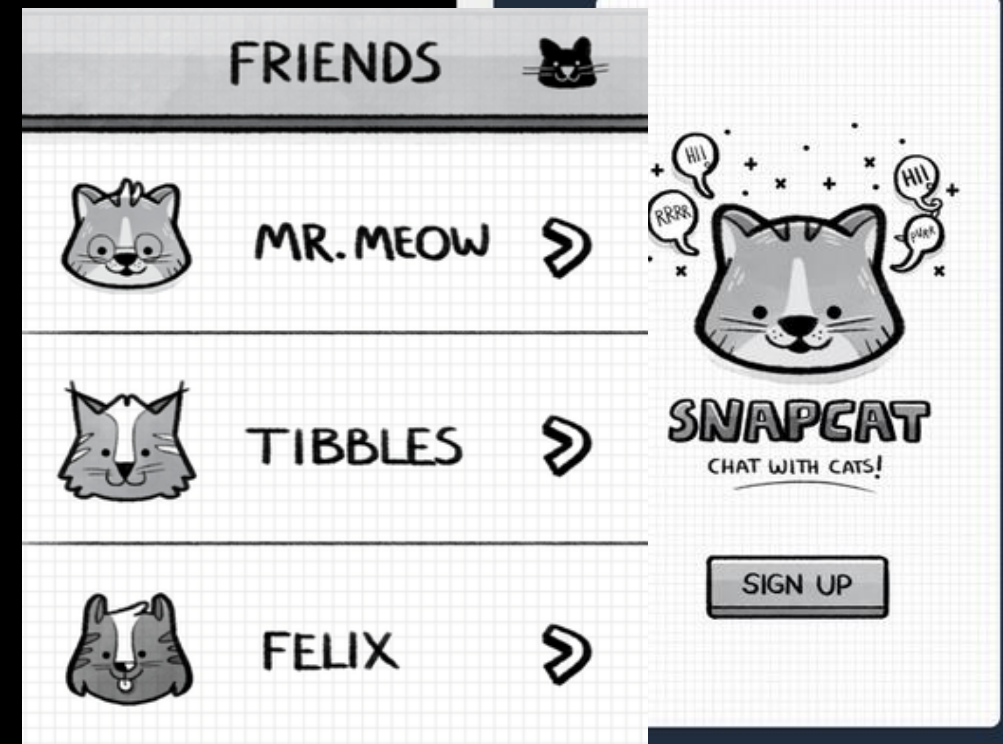
Uizard Autodesigner

You bring the ideas, AI does the rest

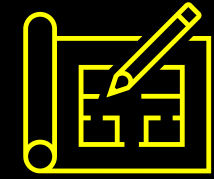


Bg. Prototype Design: POP App.

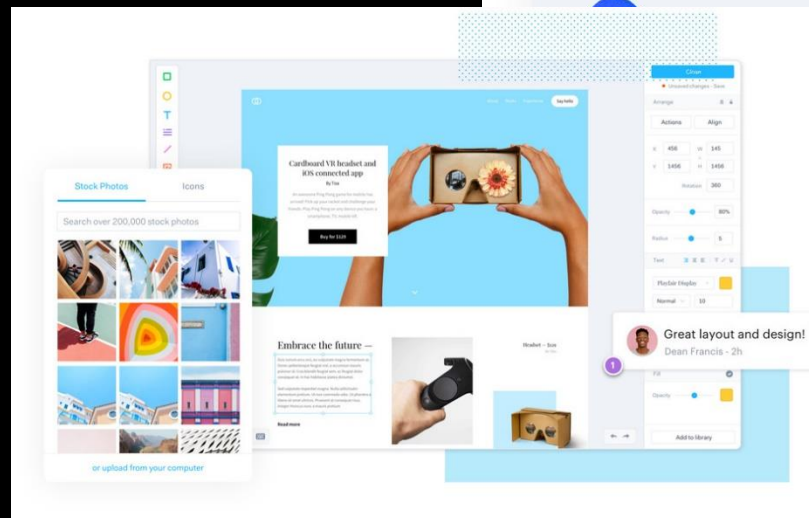
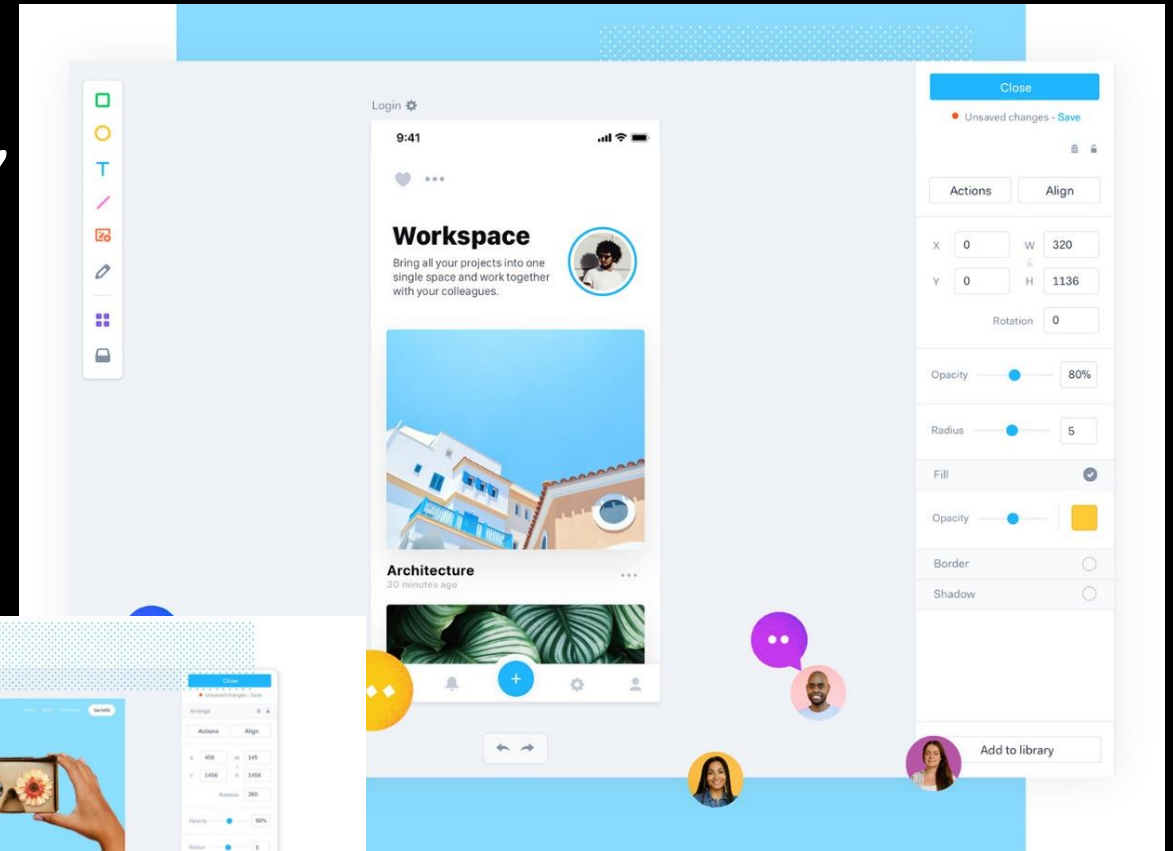
- For initial **Mobile App. Prototype design**, you could use the POP 'Prototyping on Paper' App.
- POP allows you to turn any drawing/sketch or image into an interactive prototype.
- Key links:
 1. POP App: for '[Prototyping on Paper](#)'.
 - Bg. [15sec Video Overview](#).



Bg. Prototype Design: Marvel App



- For initial **Rapid App. Prototyping design**, you could try the Marvel App., that allows you to individually or collaboratively create interactive prototype Mockups.
- <https://marvelapp.com/>.
 - Bg. 6min Video Overview: [click here](#).



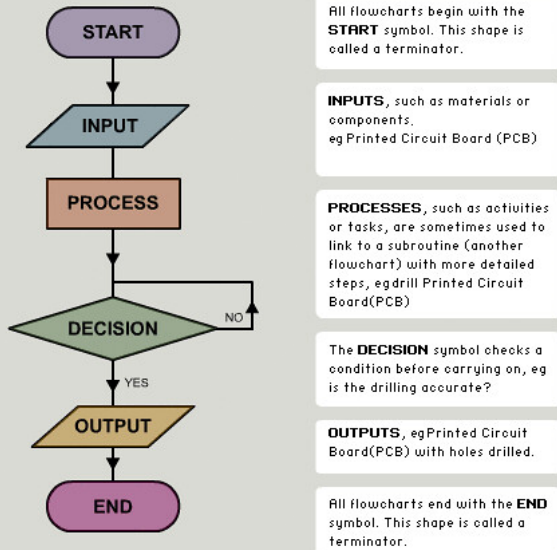
Adobe Xd



- **Adobe XD (Adobe eXperience Design)** is a vector-based UX User eXperience design tool for web and mobile apps.
- Adobe XD allows you to create wireframes and click-through prototypes / mockups.
- As a UWTSD Computing student, you are able to use the Adobe Creative Cloud suite including Adobe XD for free.
 - [Click here](#) for more information:
Else: [65% off for UK students*](#). *At time of writing.
- For more information click below for the:
 - [Adobe XD Homepage & Overview.](#)
 - [Adobe XD features.](#)
 - And here for [Video overview and tutorials.](#)

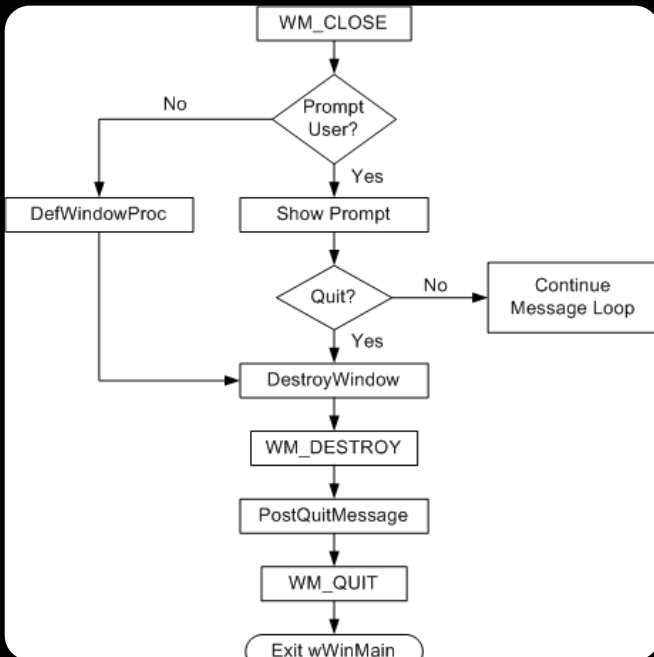


System flowchart symbols



Flow charts

- A flowchart is a diagram that shows the intended/existing sequence of functions of a program.
- Example tools for creating Flowchart & UML diagrams:
 - [Diagrams.net](https://diagrams.net) (previously Draw.io)
 - [Lucidchart - Online flowchart maker](https://lucidchart.com)
 - [Smartdraw Flowchart Maker \(Free\)](https://smartdraw.com)
 - [ClickCharts Flowchart Software](https://clickcharts.com)
 - Drawing tools in Office Apps. such as Word can also be used.
 - Bg. For further information:
 - [What is a Flowchart | Lucidchart](https://lucidchart.com/what-is-a-flowchart)
 - [What is a Flowchart and its Types?](https://lucidchart.com/what-is-a-flowchart-and-its-types)



UML diagrams

- The Unified Modeling Language (UML) are diagrams used in object-oriented software design to outline and represent the objects/features, structure and behaviour of an Application.

- For further information:

- [What is a UML Diagram?.](#)

- [UML Diagrams.](#)

- [UML diagrams: What are they and how to use them.](#)

- Video introduction to a UML Class diagram:

- [UML Class Diagram Tutorial.](#)

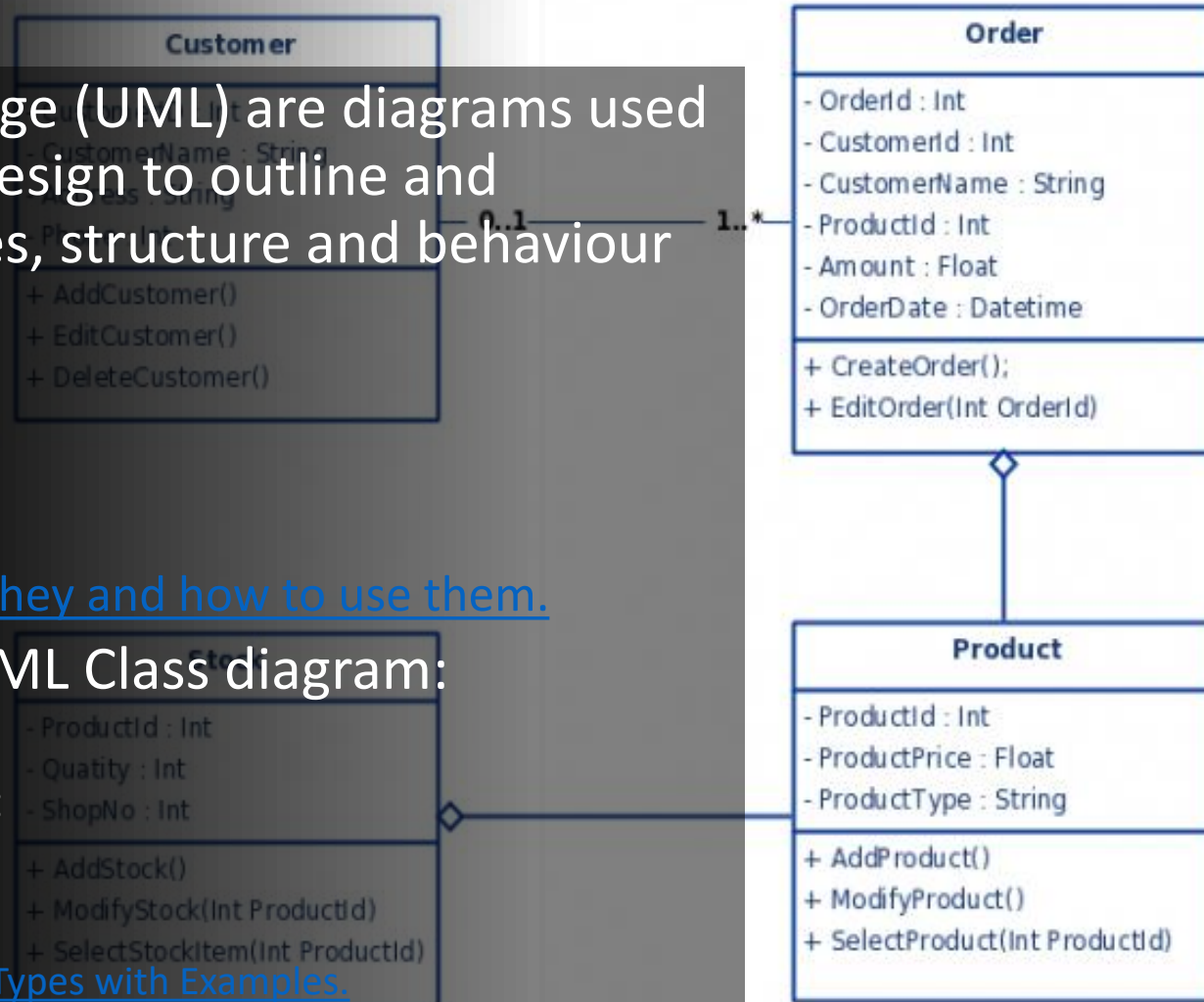
- UML Diagram types, click here:

- [UML Diagram Types Guide.](#)

- [UML Diagram Types.](#)

- [Learn about UML Diagram Types with Examples.](#)

Class Diagram for Order Processing System



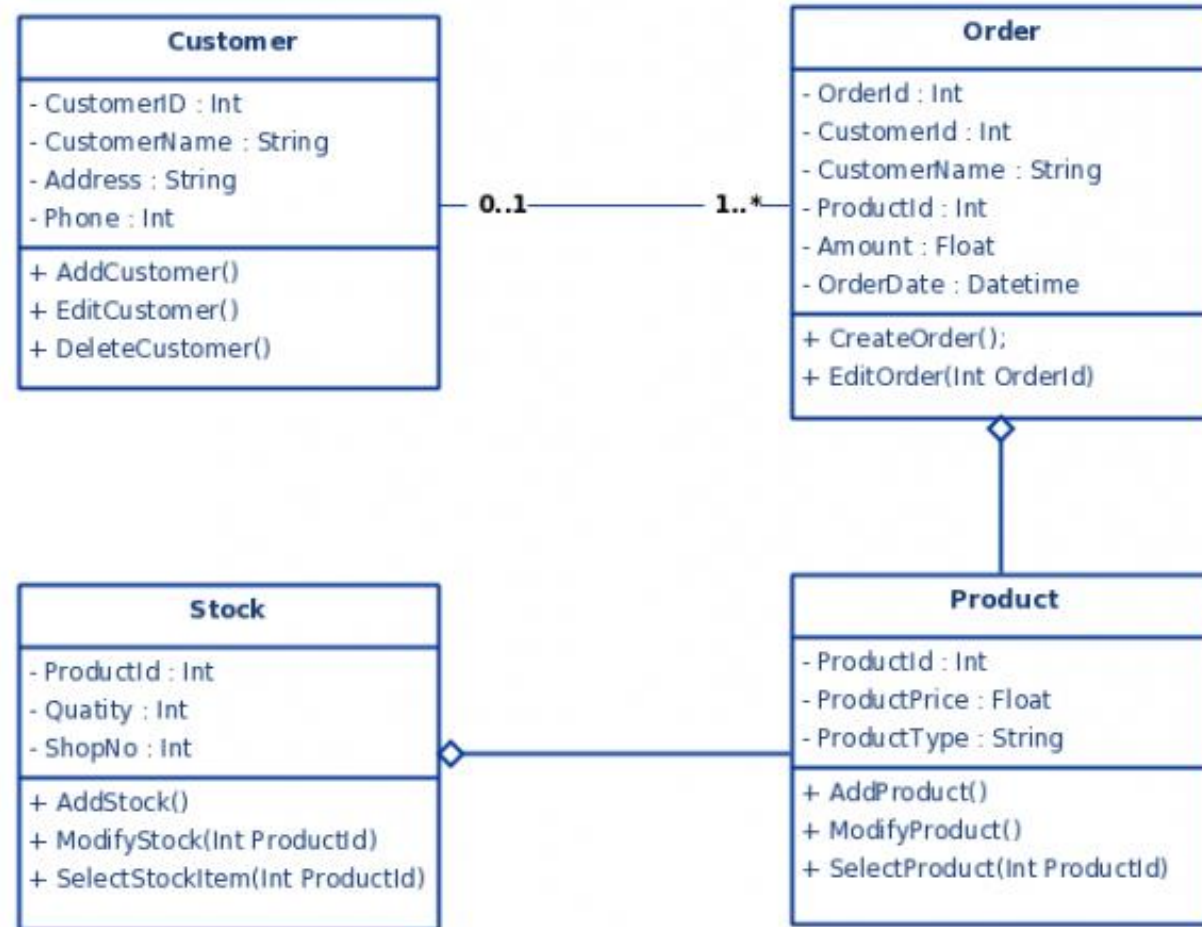
Bg. UML

Class diagram

Shows the Class/objects of a system. Each Class has a:

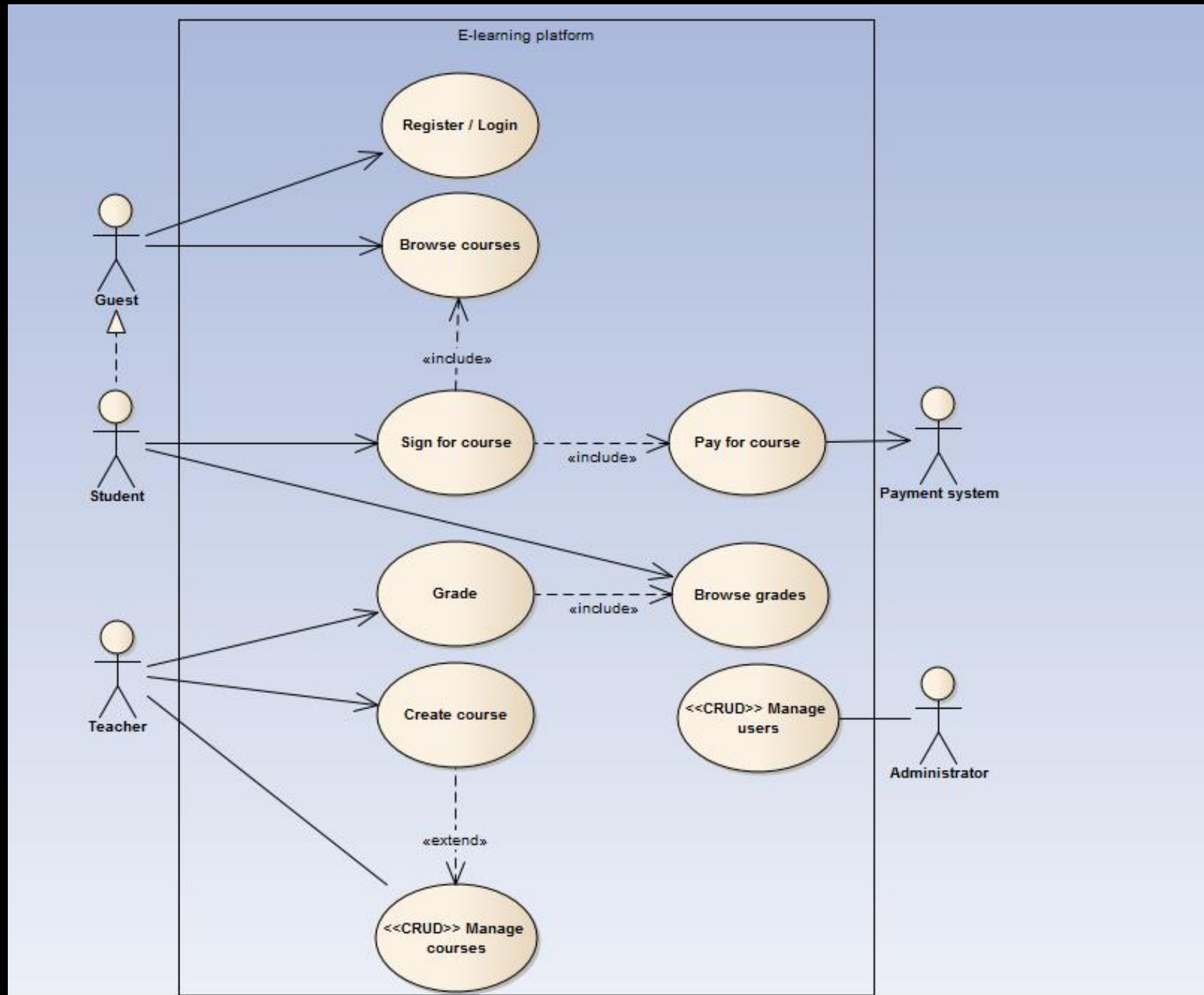
- * **Name**
- * **Attributes**/variables and
- * **Methods**/functions.

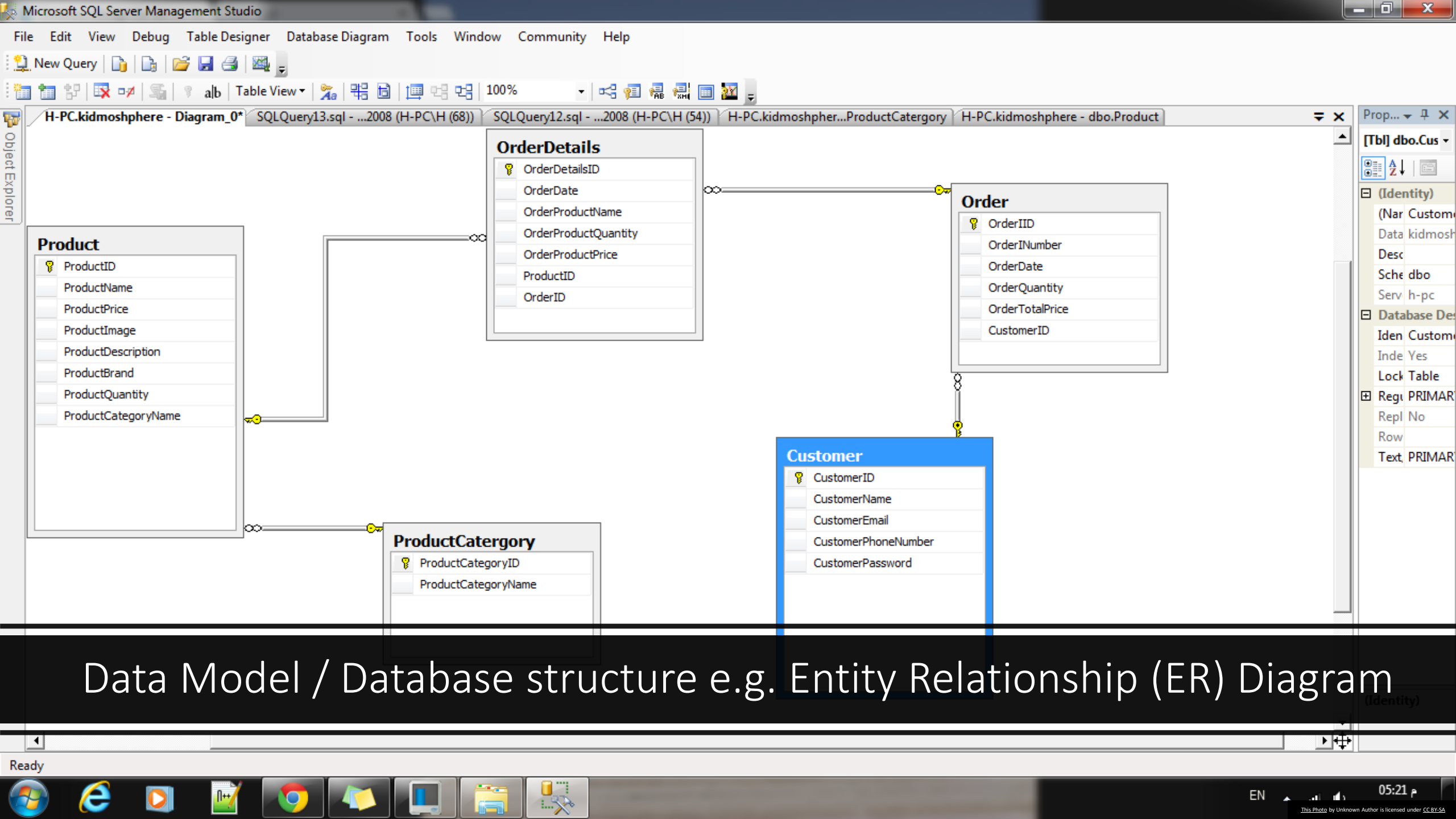
Class Diagram for Order Processing System



Bg. UML Use Case diagram

Shows the interaction between the Actors/Users and the system.

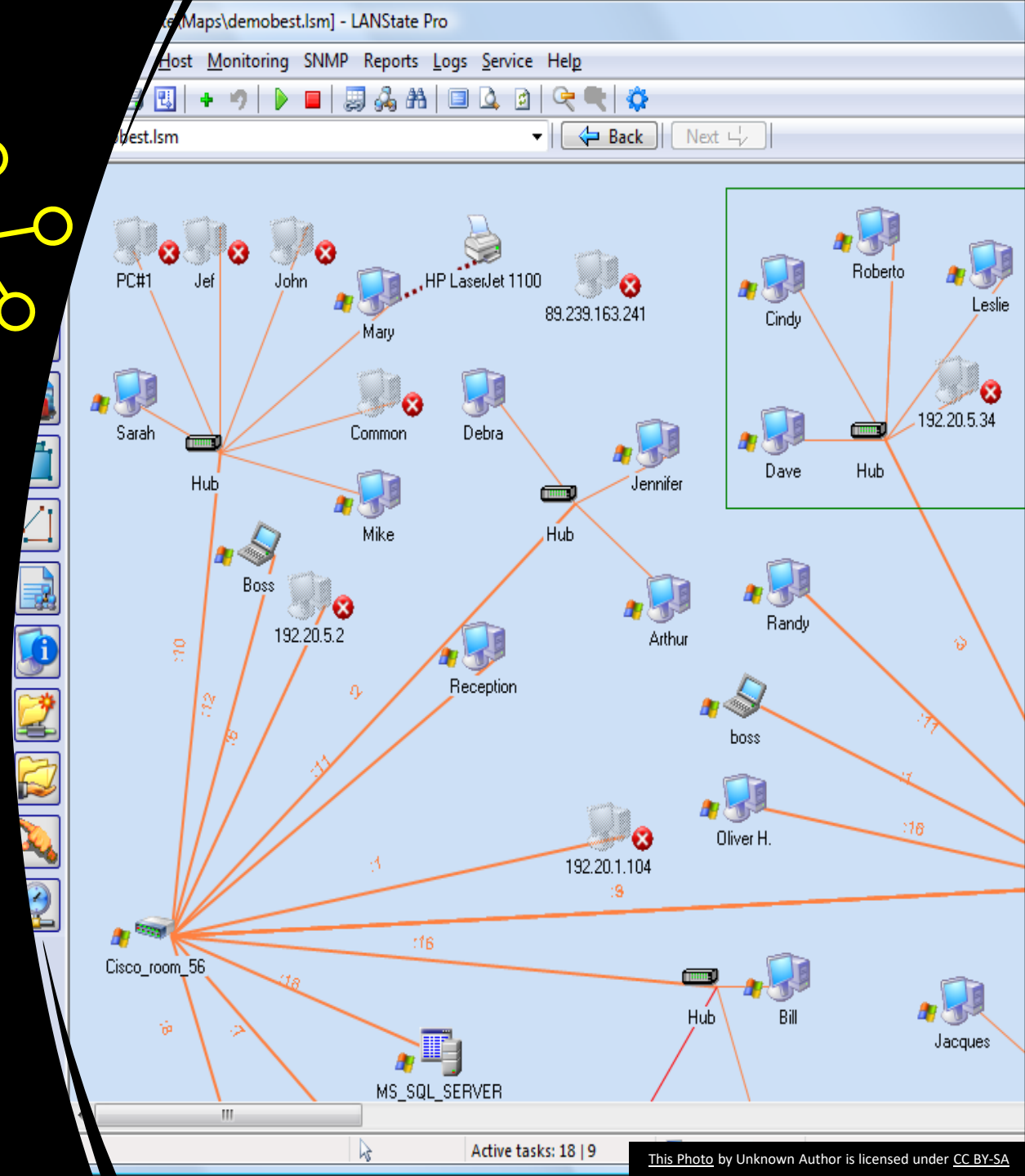
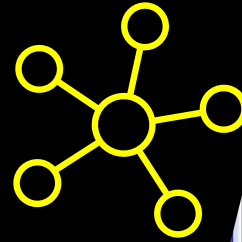




Data Model / Database structure e.g. Entity Relationship (ER) Diagram

Network & CyberSec. Mapping Monitoring & Analysis

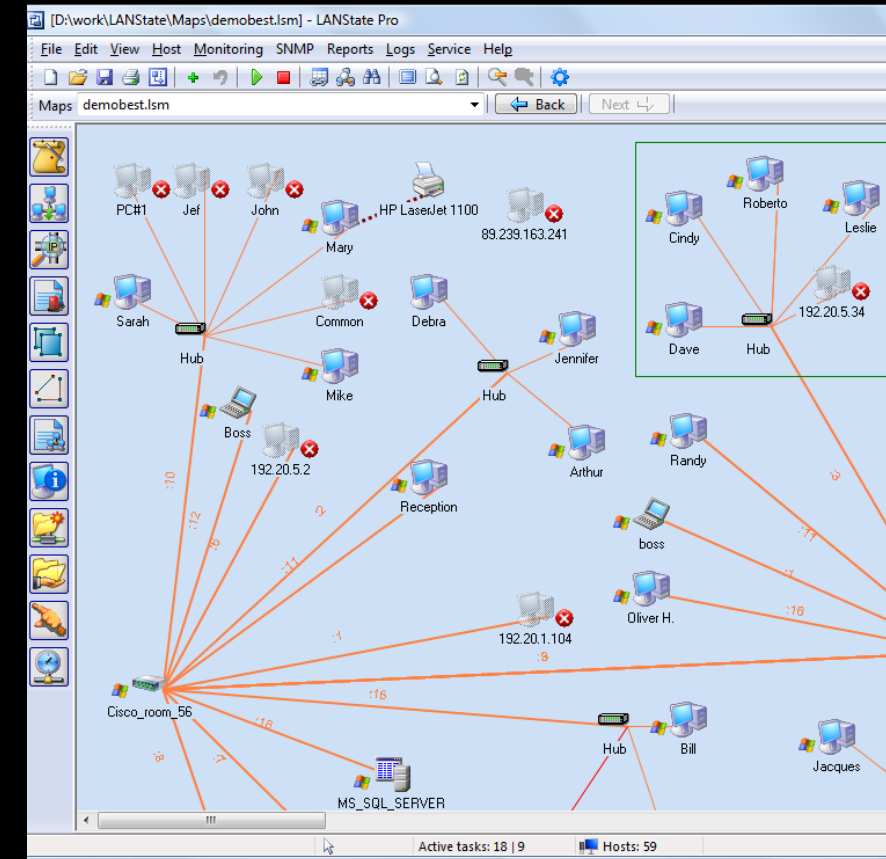
- You can create network maps or diagrams during the design phase (+ during the implementation and testing phases too).
- Or Cyber Security charts maps or diagrams using network and security mapping and monitoring software.



Network & CyberSec. Mapping, Monitoring & Analysis tools



- Many tools are available to help you design a solution, such as Network Topology creators, Network Performance Monitors, Pen/Penetration Testing tools, Network & Server Security tools.
- For Network Topology & Mapping software:
 - [Top Network Mapping Tools.](#)
 - [Best Network Mapping & Topology Software \(2022\)](#)
 - [Visual Paradigm Online - Network Diagram software.](#)
- For Network Security & Analysis software:
 - [Network Security Analysis & Monitoring Software.](#)
 - [14 Best Network Security Tools For Safer Environments.](#)
 - [13 Best Network Security Auditing Tools.](#)



Product Design Specs (PDS)

- **Product design specifications** are suitable for **hardware / electronic** prototypes, and usually include a diagram/image that documents the prototype's proposed characteristics, component list and dimensions.

PRODUCT SPECIFICATION		KELVINATOR
VIAN	FEATURES	
	1. Wash Capacity – 6.2kg 2. Spin Capacity – 4.5kg 3. Rust free plastic body 4. Single layer body 5. Transparent window 6. Normal Pulsator 7. 2 Wash Programs 8. Auto Soak 9. Buzzer 10. Net Filter 11. Deco Knob 12. Spin Shower Rinsing 13. 3 Water Level Indicator 14. Castor	
	Technical Specification	
Graphics may vary in actual mc.		
Model Name : VIAN		
Nomenclature : WM KS62VNGL-CME		
Color : GRAY LIGHT		
Wash Capacity : 6.2 Kg		
1. Machine dimension : 760x450x885 (W x D x H)mm 2. Carton dimension : 778x475x918 (W x D x H)mm 3. Rated Power : Wash - 340W, Spin – 135W 4. Voltage : 230V / 50Hz 5. Weight: Net – 19kg, Gross – 23kg		

Example: Product Design & Specification

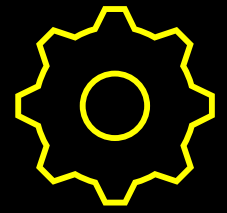


Illustration:

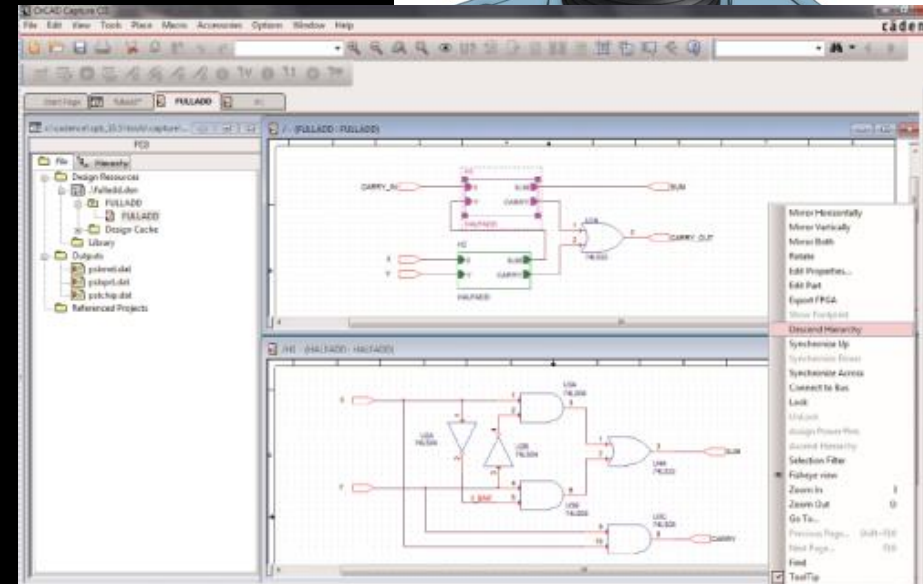
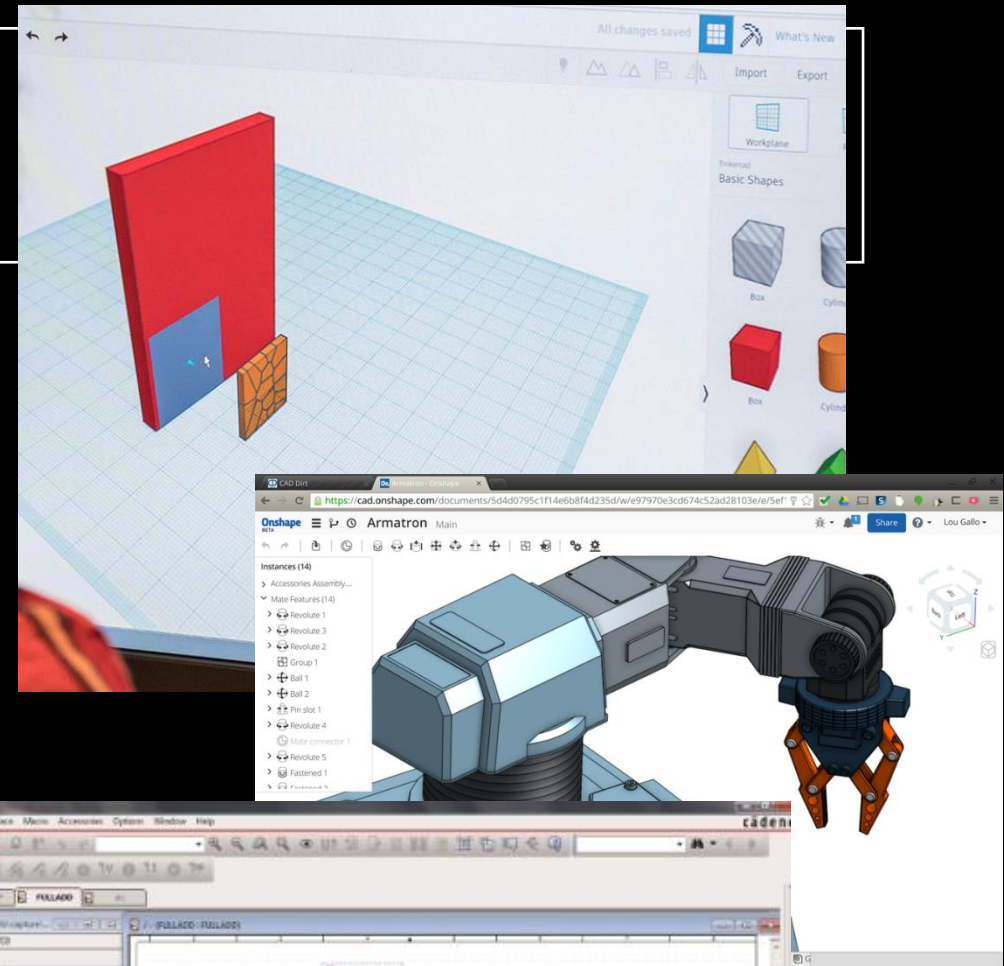
Specification:



Specification	
Processor	Intel Core 2 Duo Processor
Front Side Bus	Front Side Bus 1066 MHz
Memory	SODIMMDDR800 4GHz
Core Logic	Intel GM45/ ICHM (North / South)
Video	Intel Graphics Media Accelerator 4500MHD
Audio	Realtek ALC888 Software High Definition Audio CODEC
LAN	Built-in high speed Ethernet 10/100/1000 Mbps LAN Controller
Firewire	Built-in OHCI 1394a 1.1 compliance 400Mbps transfer rate
USB 2.0	Dynamic connection support to USB 2.0 or USB 1.1 devices
SATA	2.5' hard disk drive & optical disk drive
Hardware Monitor	Processor temperature; Voltage; CPU fan speed
Power Management	ACPI 1.0b compliance and OS direct power management

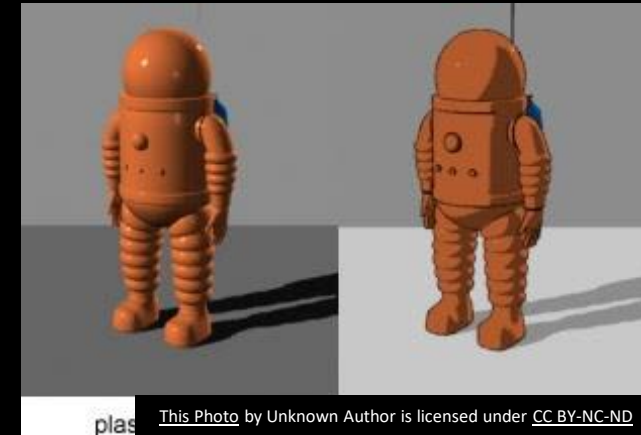
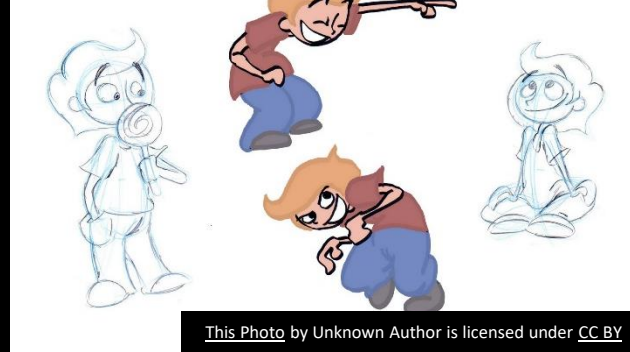
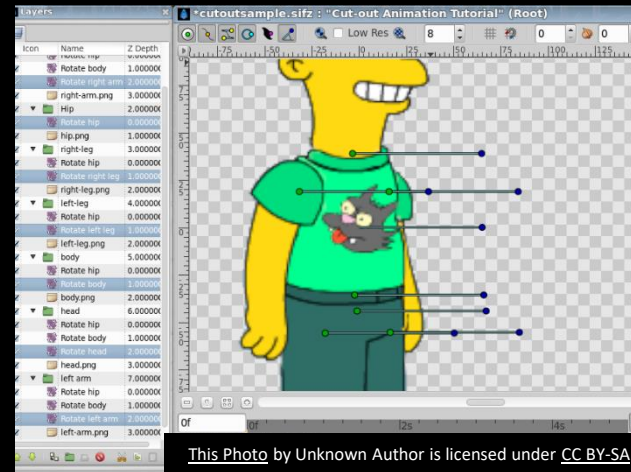
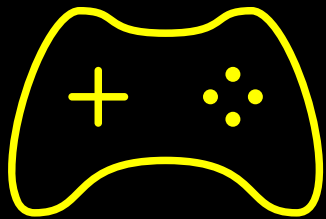
Hardware Prototype design.

- For initial **Hardware Prototype design**, you could use CAD (E.G. AutoCAD by Autodesk, SmartDraw or other free alternative), Adobe Illustrator or Photoshop.
- For electronics you could use '[Proteus Design Suite](#)' to create automate and test electronic circuit board schematics.



Games Des. Dev. &/or Animation

- If your project is Game or Animation based, you can include initial design work such as character design, sketches, illustrations, models, storyboards.

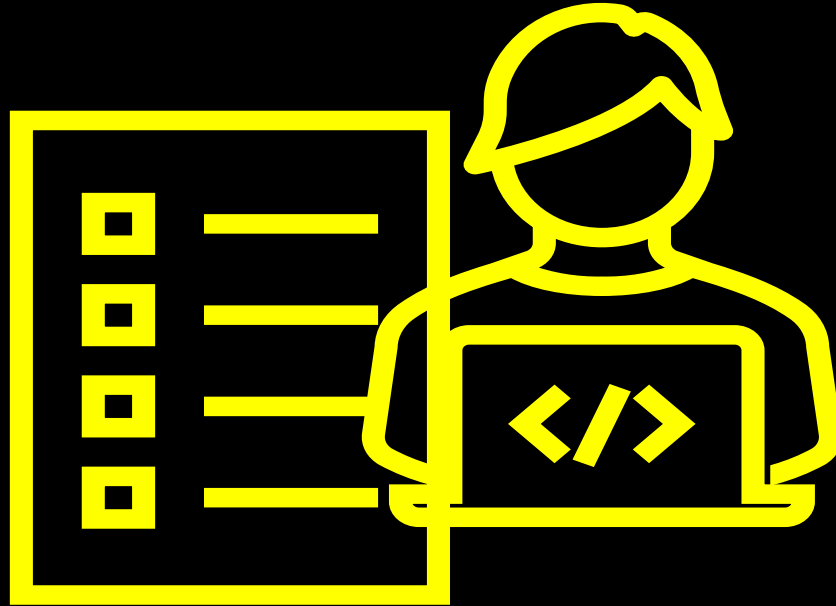


Keep it Simple: Design & System Modelling



- Keep the Design docs simple.
- For this module you do not need to go into detail with your Design planning & documentation.
- Keep it **Clean Lean & Simple**.
- The Design documents produced by different students will vary significantly and depend on your project area. You will all have different design documentation. And that's ok.

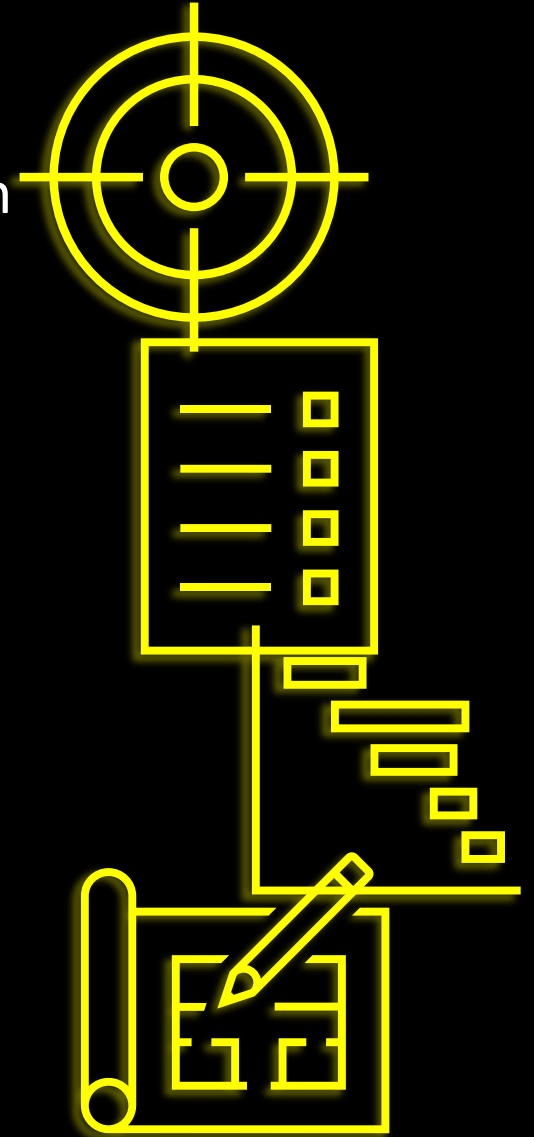
Project Manager



Project Manager: Hard & Soft Skills

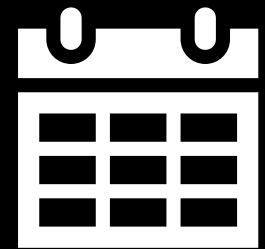
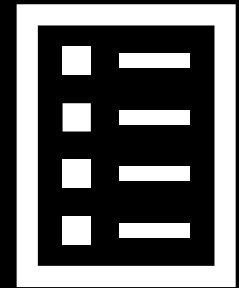
Your project will help you to develop both hard and soft skills. Both are equally required. Such as:

- **Hard skills:** refer to more objective practical and technical skills you will develop while undertaking your project e.g. Computing technical skills for design and development/implementation and other digital skills (e.g. using Office software, Project management/Gantt chart software).
- **Soft skills:** refer to more subjective personal professional and social skills you develop while undertaking your project e.g. organisation, planning & project management skills, time management, motivation, resilience, communication skills, professional legal and ethical considerations.
- Developing the soft skills are more important for a Project Manager, for successful Project Management.

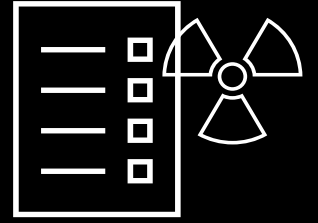


Project constraints

- You will be the '**Project Manager**' of your Project.
- Every project has constraints and finite resources, and every Project Manager needs to work within these constraints to produce the best possible outcome.
- To complete your project **on time** and with **limited resources**, will take project planning, resourcefulness, organisation and time management.
- You will be supported by a Project supervisor who can assist you. But ultimately you will need to manage your time and resources to complete your project within the time frame.



Project Constraints



Project constraints are **limiting factors and risks** that may occur during your project that you need to be aware of and work with. Such **Project constraints** include:

- **Limitations:** Limitations include time and resources available and current skill level. Being aware of these will help you to set the scope of your project and to develop your skills in project related areas.
- **Risks:** A risk is an uncertain event that may or may not occur during a project, but is worth planning for to prevent disruption in advance. E.g. in the list below a minus '-' = **Risk** & a plus '+' = a **Contingency**
 - **Running out of time** + Project Planning / Gantt Chart.
 - **Loss of data** + Apply the rule of 3: Back-up work in 3 different places.
 - **Hardware failure** + Back-ups, Tech Support, Where to acquire or use a back-up device e.g. from Uni/Family/friend or ring-fence a fund to purchase a new one.

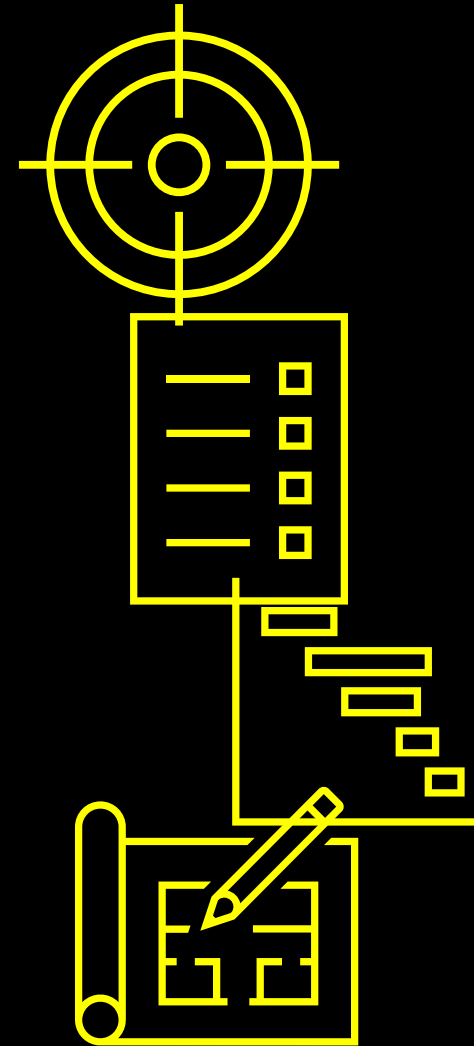
Project Risks & Contingencies

- Project constraints, risks and challenges can be outlined using a simple table, like the one below. This format can be used to answer Task 3c. Example below:

Risk no.	Risks	Impact	Contingencies
1.	Hardware failure & loss of data	High	The rule of 3 will be applied from the outset, where project/Uni work will be backed-up in 3 different places, including a secure reliable cloud facility using Onedrive and Google drive.
2.	Skill set	High	Relevant resources will be found and used to develop the required skills to complete the project, prior-to, during and throughout the project process.
3.	Time & Time management.	Medium	Running out of time to complete the project is always a risk. A Gantt chart will be used to outline key stages and tasks that need to be completed. This will be followed and updated throughout the project to ensure I stay organised, focused, to complete the project on time.
4.	Gold plating	Medium	Focusing on and adding features that are not directly related to the Aim & Objectives of the project. Reducing the time to finish the key high priority parts of the project. To avoid this risk a prioritised Requirements specification and Gantt chart will be used to help focus on the key features to be implemented. If there is time left at the end of the project, lower priority and desired / out-of-scope ideas can be added after.

Conclusion

1. Project Planning
 - inc. Project Scope
 - Inc. Project Stages & Schedule
2. Project Planning & Management
3. Project Development Methods
4. Project Design Methods
5. Project Manager & Risk Management





The End.

Questions.

Q&A.

