# 3. Website Report

UCL

3

UCL ENGINEERING
Change the world

# Website Structure

- Home
- Requirements
- Research
- Algorithms (if applicable)
- UI Design (if applicable)
- System Design
- Implementation
- Testing
- Evaluation
- Appendices
  - User manual and deployment manual
  - GDPR and Privacy of Data
  - Development Blog – External Link on a public site (**this should be updated every two weeks**)
  - Monthly Video

# Website Submission

- You can use **any tools or frameworks** to develop your report website but the submitted website should be a static website that **only includes the HTML, CSS, JavaScript files.**
- **The examiners should not be required to install anything and can open the index.html file to navigate all the pages.**
- **Please ensure all the links work** – especially for the index.html page **to navigate all of your pages**. We won't open the html files that are not linked.

# Home

- Project title

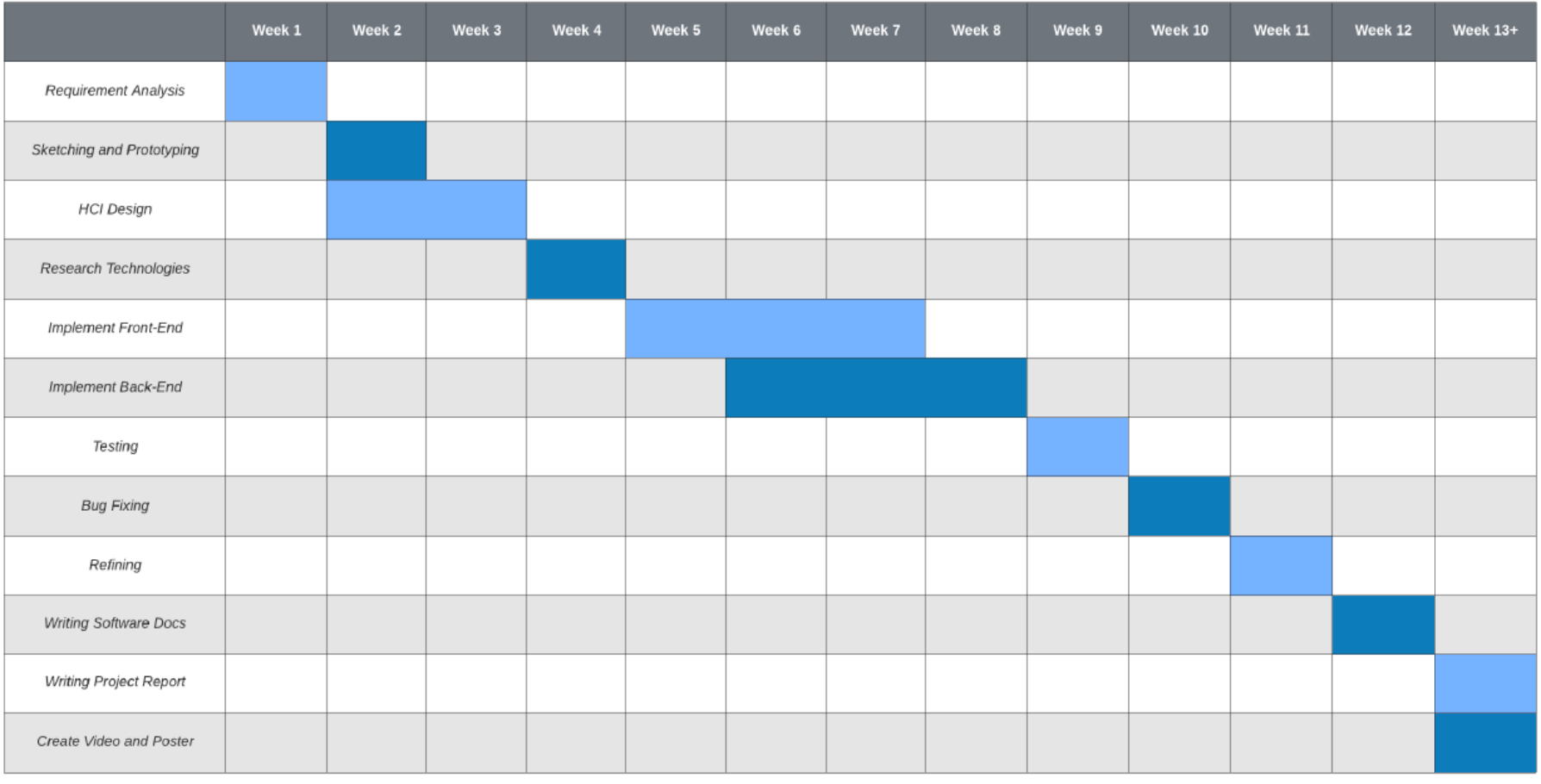
- An abstract should include three paragraphs and the following contents
  - Problem statement
  - Your solution
  - Achievement and impact

- A 8 minutes video to introduce the project and go through the finished functionalities

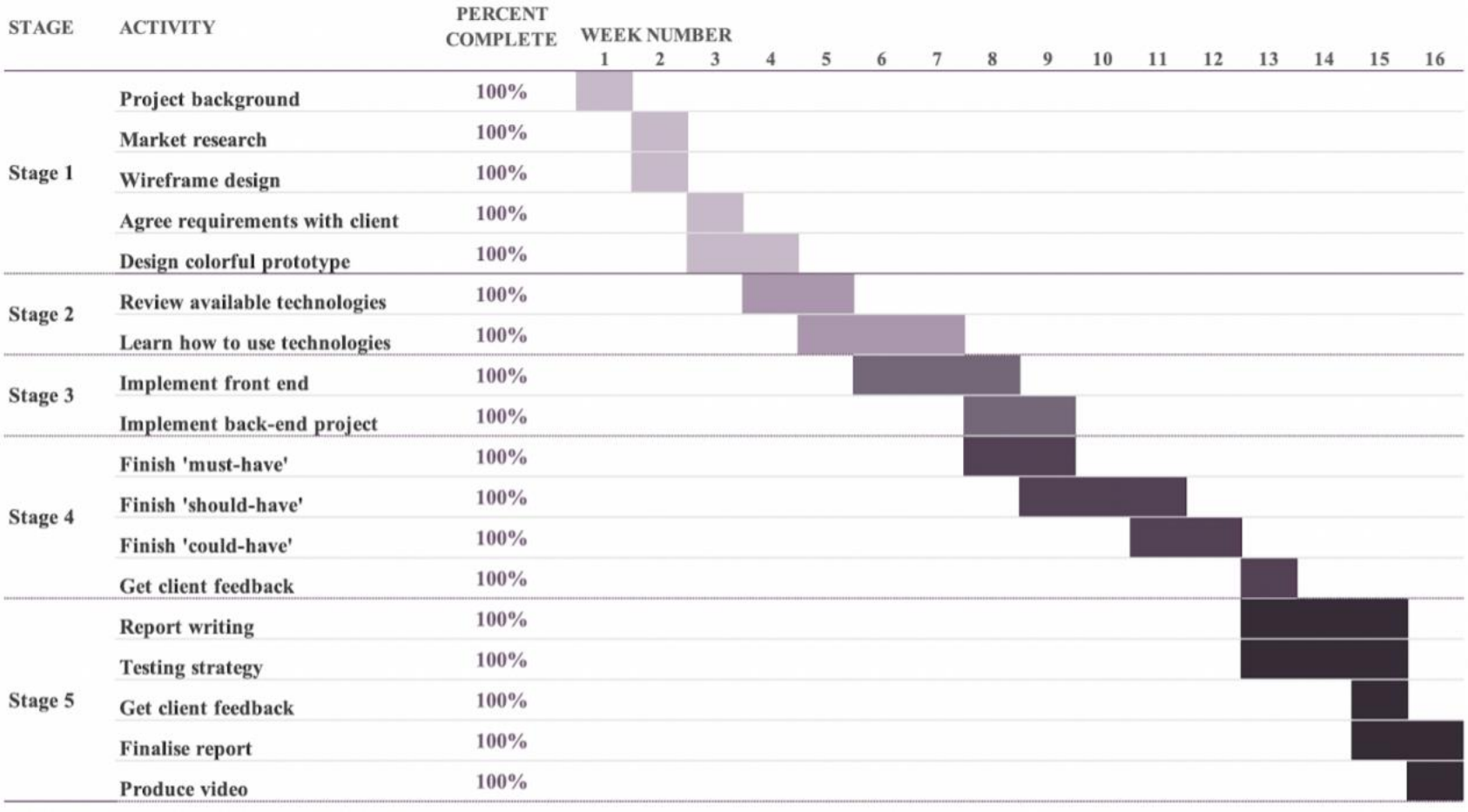
- The development team
  - (including the team photos where possible) name, email, role or main contribution of each member. The roles include client liaison, UI design, researcher, programmer, report editor, tester.
  - You are welcome to include interesting background about yourself, as well as your LinkedIn profiles if you wish.

- Project management
  - Gantt chart (from October 21 2024 to March 28 2025)

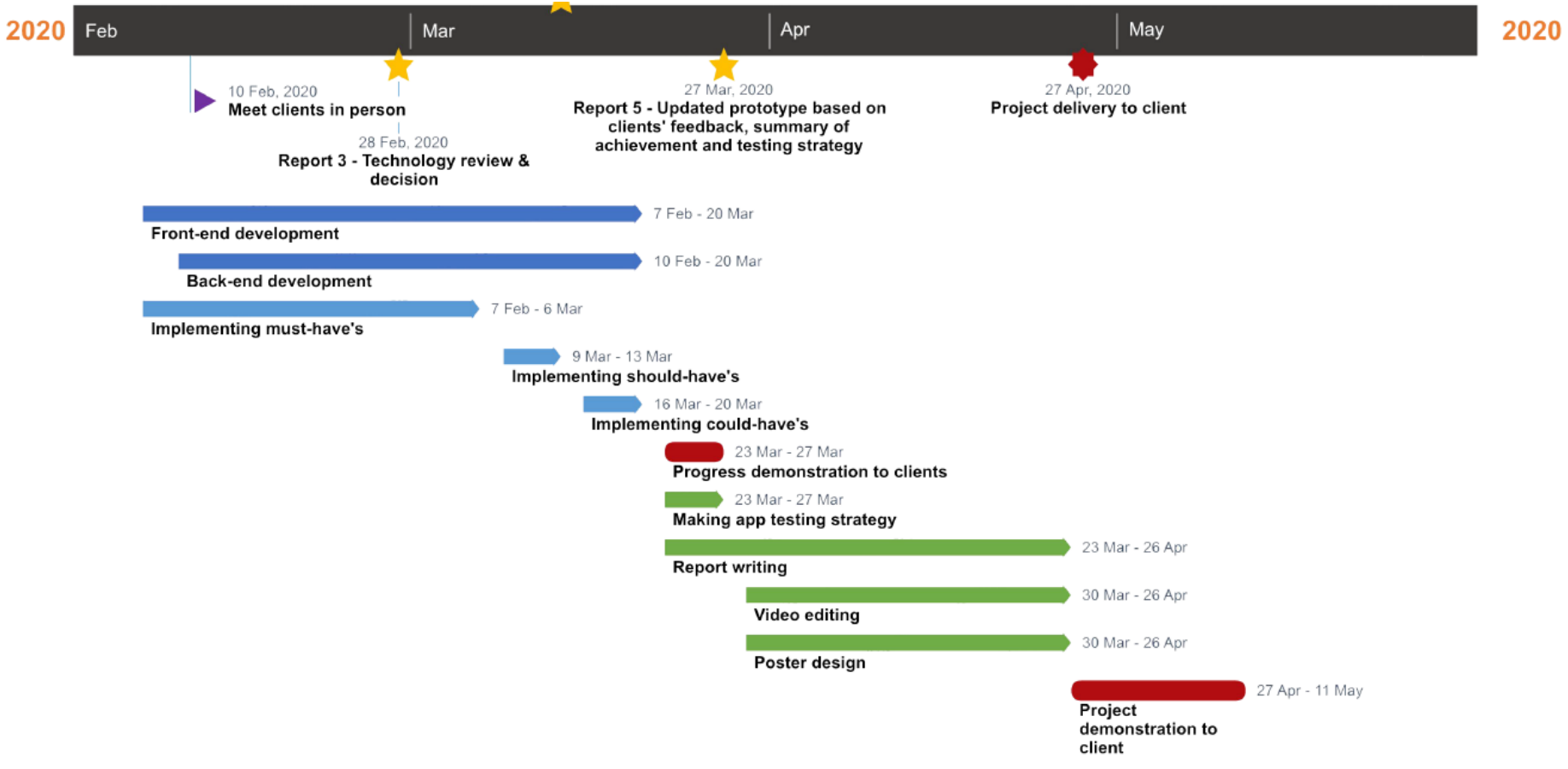
# Gantt Chart Example 1



# Gantt Chart Example 2



# Gantt Chart Example 3



# Requirements

- Partner Introduction and Project background
- Project Goals
- Requirement gathering
  - How did you collect the requirements?
  - Did you design any survey? How did you analyse the survey data?
- Personas
  - Typical users of your project
- Use cases (if applicable)
  - Use case diagram
  - List of use cases
- MoSCoW requirement list (Functional and non-functional)
  - A table for function requirements
  - Another table for non-functional requirements: https://en.wikipedia.org/wiki/Non-functional_requirement. For example, performance, security, usability, open source, maintainability, extensibility, etc.

# Persona Example 1

## PERSONA 1

NAME Rachel Adam
OCCUPATION Student
AGE 16
STATUS Amateur

"I would like to improve my bow arm technique so I can produce better articulated sound"

### MOTIVATION

Rachel has just finished her GCSEs and has decided to apply for music conservatoire to pursue a professional musician career.

### BACKGROUND

She is just like very other teenage that her smartphone never leave her sights. She has used mobile applications to aid her revision during GCSEs.

### GOAL

- Keep a daily diary to monitor progress
- Improve sound production through right hand bow techniques

# Persona Example 2

Clinician

**Name:** James McMillan
**Age**: 32
**Occupation:** Clinician
**Bio**: James is a clinician at NHS. He works with patients directly and has experience working with common medical software.

Researcher

**Name**: Claire Jones
**Age**: 38
**Occupation**: Researcher at UCL Institute of Child Health
**Bio:** Claire works with various data gathered during the project. She is mostly interested in common trends and NOT in particular patients

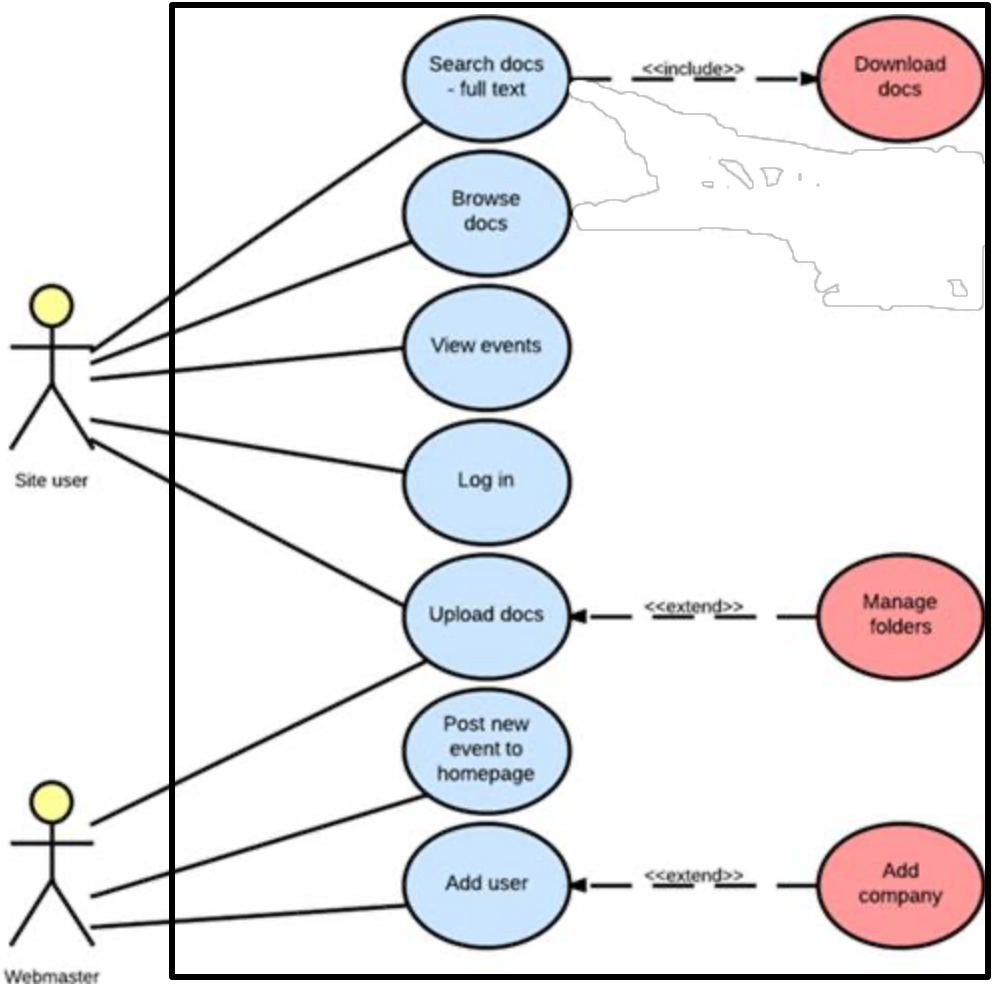
Parent

**Name**: Anthony Paradzinski
**Age**: 42
**Occupation**: Works as a professor and is also a Mike's father.
**Bio**: Anthony wants to know how his son Mike, who has CF is feeling. Anthony needs to be able to ensure Mike does all the prescribed amount of daily exercises

Patient

**Name**: Mike Brown
**Age**: 10
**Occupation**: Student
**Bio**: Mike is a patient at UCLH and needs to do daily breathing exercises. With Fizzyo, he wants to see how he is performing against the prescription and also see what game achievements he unlocked.

# Use Case Diagram Example



# Use Case List

## A list of use cases

| ID | Use Case For User |
|---|---|
| UCU1 | Viewing the News section |
| UCU2 | Viewing the Events section |
| **UCU3** | **Searching for Mobile applications** |
| **UCU4** | **Contacting Mr Yun Fu** |
| UCU5 | Finding the Location of UCL |
| UCU6 | Finding the phone number of IXN |
| UCU7 | Searching for Web applications |
| **ID** | **Use Case for Administrator** |
| UCA1 | Posting an Event |
| UCA2 | **Posting a Project** |
| UCA3 | Updating the News section |

## Use case description

| Use Case | |
|---|---|
| ID | UCA2 |
| Actor | Administrator |
| Description | Posting a Project |
| Main Flow | 1. System displays an administrator login view.<br>2. Administrator logs in.<br>3. System offers list of sections<br>4. Administrator selects WP selection.<br>5. System prompts for Title and Description of new project, video upload, poster photo and site image.<br>6. Administrator enters information and submits.<br>7. System displays a confirmation response. |
| Result | New Project Posted |

# MoSCoW requirement list

- **Functional** requirements

| ID | Requirements | Priority |
|---|---|---|
| 1 | | Must have |
| 2 | | Should have |
| 3 | | Could have |

- **Non-functional** requirements

| ID | Requirements | Priority |
|---|---|---|
| 1 | | Must have |
| 2 | | Should have |
| 3 | | Could have |

# Research

- Related Projects Review
  - Review at least 1 existing similar project if applicable
  - For each of the similar projects, please list the project name, main features, and what you can learn from this existing application.
- Technology Review
  - Please compare the possible solutions, describe what you choose, and explain why.
  - Please compare the possible devices (if applicable), describe what you choose, and explain why.
  - Please compare the possible algorithms (if applicable), describe what you choose, and explain why.
  - Please compare the possible programming languages, frameworks, libraries, APIs, describe what you choose, and explain why.

- A summary of your technical decisions

- References
  - IEEE style: reference definition needs the number

    "Several recent studies [1, 4, 15, 22] have suggested that. . . ."

    [1] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 2nd ed. Reading, MA: Addison Wesley, 2003. [E-book] Available: Safari e-book.

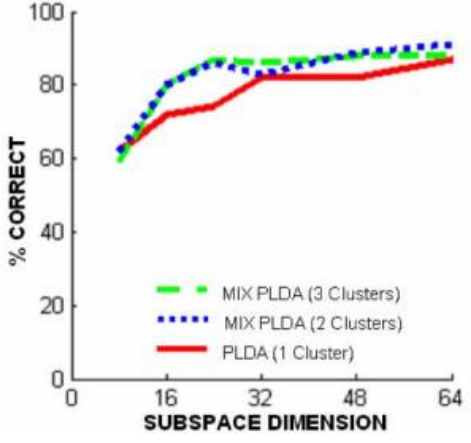
  - IEEE citation reference definition
    - https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf

# Algorithms (if applicable)

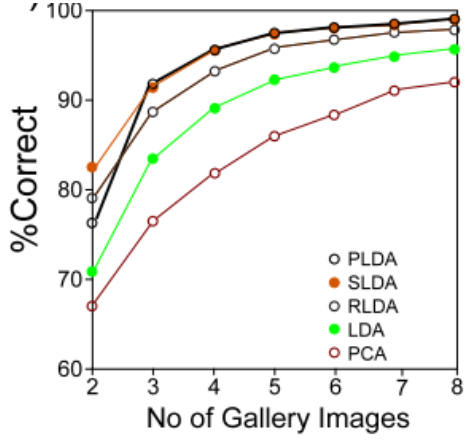
- Models
  - Describe the key idea of your chosen algorithms
- Data
  - Dataset
  - Data Preprocess (if applicable)
  - Training and testing sets
- Experiments
  - Experiment design
  - Performance evaluation method (e.g. % accuracy)
  - Experiment results that should be presented by quantified values
  - Investigation of the optional Hyperparameters (if applicable)
  - Use plots or tables to show the performance comparison results
- Discussions
  - Why the algorithm fails for some test examples
  - Suggestions to improve the performance
- Conclusion
- References

# Plot and Table Example

Investigation of the optional Hyperparameters Plot Example 1




Investigation of the optional Hyperparameters Plot Example 2




ROC Curve Example

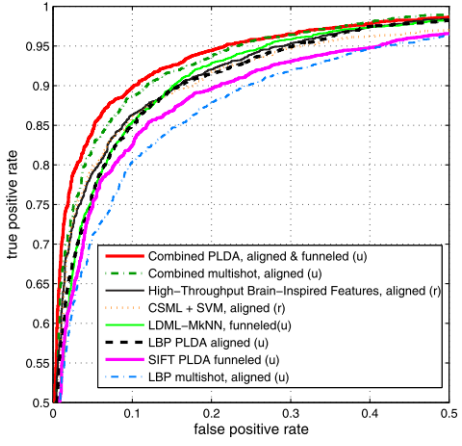



Table Example

| Method | Accuracy |
|---|---|
| Combined PLDA, aligned & funneled (u) | $0.901 \pm 0.005$ |
| Combined multishot, aligned (u) [37] | $0.895 \pm 0.005$ |
| Combined LDML–MkNN, funneled (u) [13] | $0.875 \pm 0.004$ |
| HTBI Features, aligned (r) [33] | $0.881 \pm 0.006$ |
| CSML + SVM, aligned (r) [29] | $0.880 \pm 0.004$ |
| TPLBP PLDA, aligned (u) | $0.837 \pm 0.007$ |
| LBP PLDA, aligned (u) | $0.873 \pm 0.006$ |
| LBP multishot, aligned (u) [37] | $0.851 \pm 0.006$ |
| SIFT PLDA, funneled (u) | $0.862 \pm 0.012$ |
| SIFT LDML, funneled (u) [13] | $0.832 \pm 0.004$ |

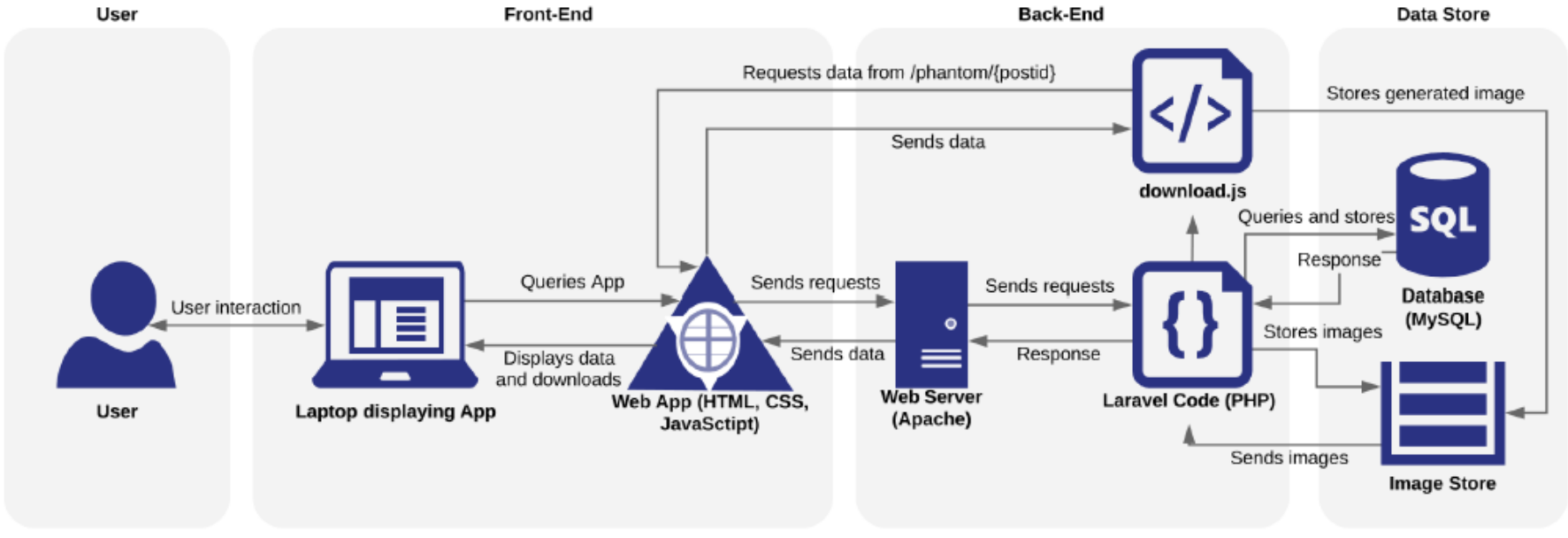
# User Interface Design (if applicable)

- Design Principles (e.g. simplicity, consistency, visibility, feedback, tolerance, etc.)
- Hand-drawn sketches
- Online interactive wireframe URL (High Fidelity wireframe e.g. with Figma)

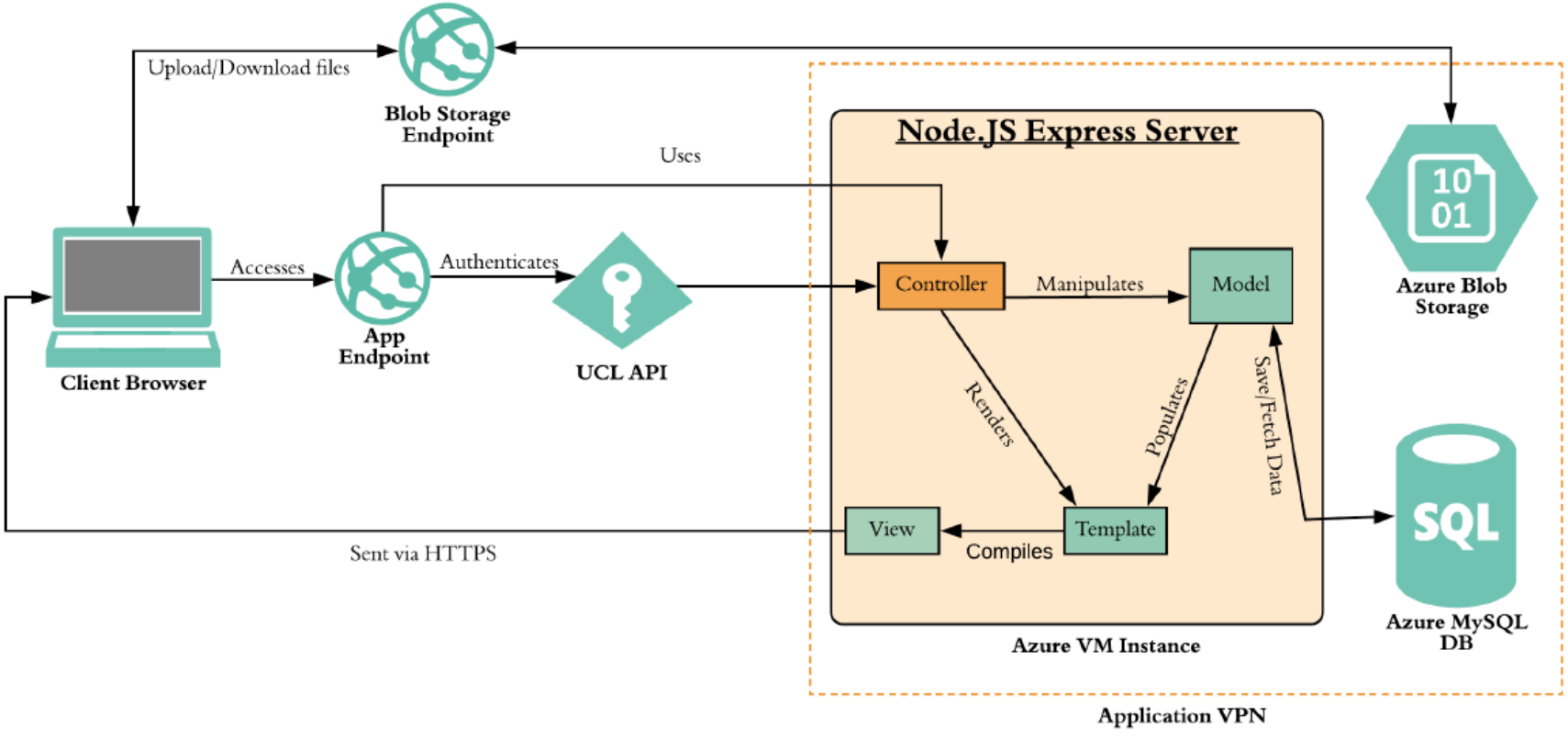

# System Design

- System architecture diagram and a brief description of each component.
- Site Map (if applicable)
- Sequence Diagrams (if applicable)
- Design Patterns (if applicable)
- Class Diagrams (if applicable)
- Data storage (if applicable)
  - Please provide the data schema (e.g. ER diagram) if you have a database
- Packages and APIs defined (if applicable)

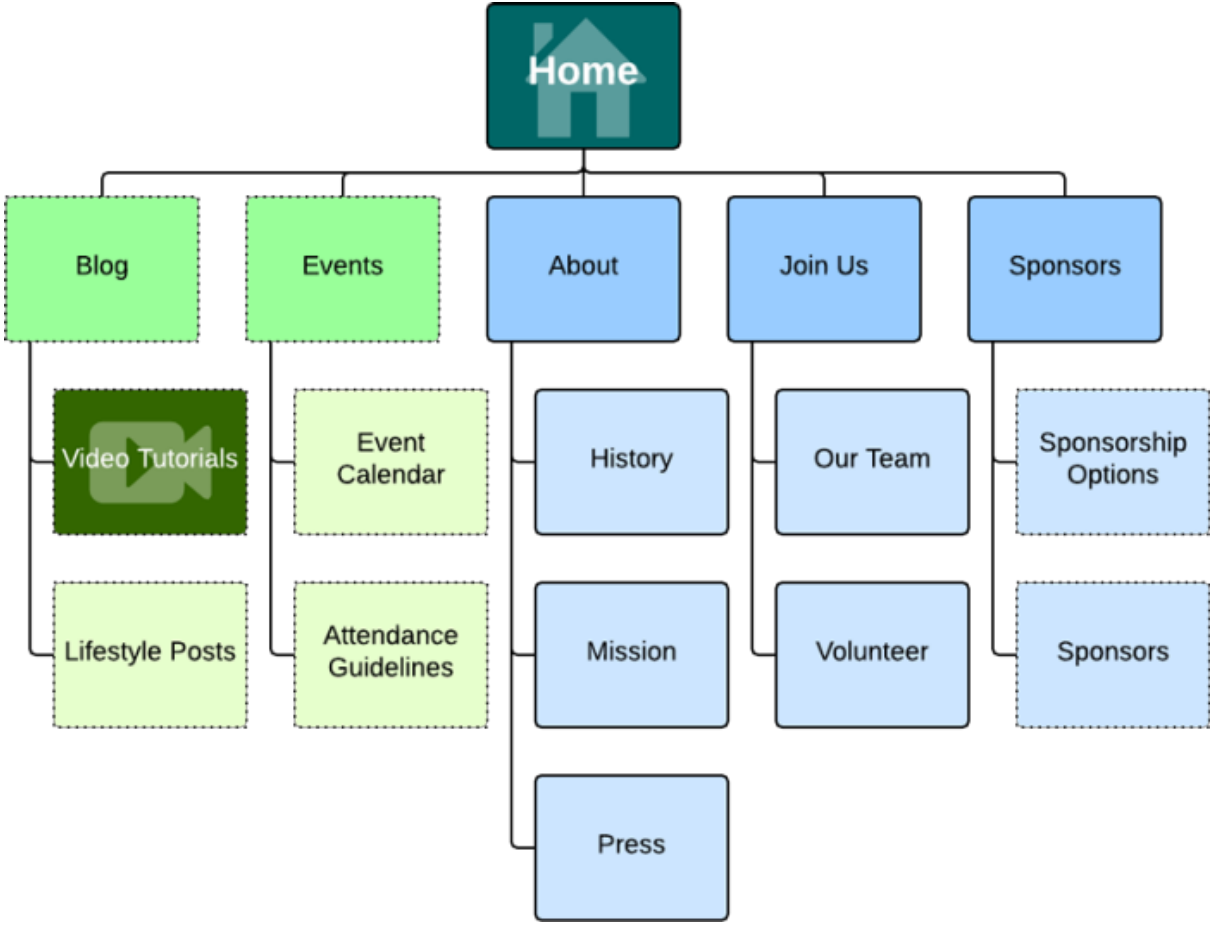
# System Architecture Example 1



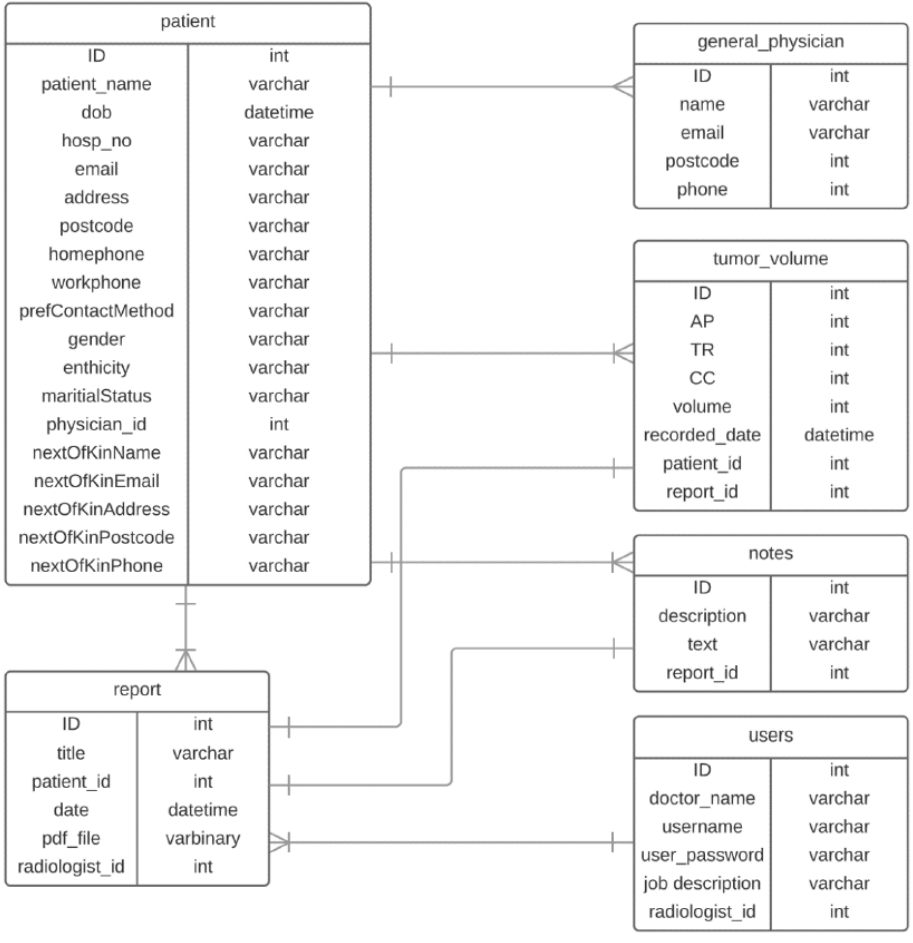
# System Architecture Example 2



# Site Map Example



# ER Diagram


patient
ID int
patient_name varchar
dob datetime
hosp_no varchar
email varchar
address varchar
postcode varchar
homephone varchar
workphone varchar
prefContactMethod varchar
gender varchar
enthicity varchar
maritialStatus varchar
physician_id int
nextOfKinName varchar
nextOfKinEmail varchar
nextOfKinAddress varchar
nextOfKinPostcode varchar
nextOfKinPhone varchar
general_physician
ID int
name varchar
email varchar
postcode int
phone int
tumor_volume
ID int
AP int
TR int
CC int
volume int
recorded_date datetime
patient_id int
report_id int
notes
ID int
description varchar
text varchar
report_id int
report
ID int
title varchar
patient_id int
date datetime
pdf_file varbinary
radiologist_id int
users
ID int
doctor_name varchar
username varchar
user_password varchar
job description varchar
radiologist_id int

# Implementation (Very Important)

- Please describe how you implement the key features
  - It is very hard for the other people to understand your source code by only reading your code. You need to provide the necessary explanation documentation to help other people to understand how you implement the key features, for example, which frameworks or plugins or libraries you used for a particular feature and how you use them.
  - Please describe one key feature in a subsection. We take a website project as an example, you can describe the implementation details of the following features: authentication, database connection, data table, trend chart, csv data import, pdf generation, notification, etc.
  - If necessary, sequence or other diagram and short code snippet can be used to help the explanation.

# Testing

- Testing Strategy
- Unit and integration testing
- Compatibility testing (if applicable)
- Responsive design testing (if applicable)
- Performance/stress testing (if applicable)
- User acceptance testing
  - Simulated testers
  - Test cases
  - Feedback from testers and project partners
- For each test, you should explain why you do this test, which test tool you use, how you conduct the test, what results you get, and your analysis or conclusion to the results.

# Evaluation

- Summary of achievements
  - An achievement table to list the MoSCoW functionalities, the completed states, and contributors
  - A list of known bugs (if applicable)
  - Individual contribution distribution table

- Critical evaluation of the project
  - User interface / user experience (if applicable)
  - Functionality
  - Stability
  - Efficiency
  - Compatibility
  - Maintainability
  - Project management

- Future work
  - How could the project be extended if you have more time

# Example of Achievement Table and Known Bug List

- Achievement table

| ID | Requirements | Priority | State | Contributors |
|---|---|---|---|---|
| 1 | | Must | ✓ | All |
| 2 | | Must | ✓ | All |
| 3 | | Should | X | John |
| 4 | | Could | X | Ali, Kai |
| **Key Functionalities (must have and should have)** | | **85%** completed | | |
| **Optional Functionalities (could have)** | | **65%** completed | | |

- Known bug list

| ID | Bug Description | Priority |
|---|---|---|
| 1 | | High |
| 2 | | Medium |
| 3 | | Low |

# Individual Contribution Table For System Artefacts

| Work packages | John | Ali | Kai |
|---|---|---|---|
| Research and Experiments | 40% | 10% | 40% |
| UI Design (if applicable) | 20% | 70% | 10% |
| Coding | 40% | 30% | 30% |
| Testing | 20% | 0 | 80% |
| Overall contribution | 30% | 34% | 36% |

# Individual Contribution Table For Website

| Work packages | John | Ali | Kai |
|---|---|---|---|
| Website Template and Setup | 0% | 66% | 34% |
| Home | 33% | 33% | 34% |
| Video | 33% | 33% | 34% |
| Requirement | 34% | 33% | 33% |
| Research | 50% | 0 | 50% |
| Algorithm (if applicable) | 20% | 10% | 70% |
| UI Design (if applicable) | 20% | 0 | 80% |
| System Design | 20% | 0 | 80% |
| Implementation | 20% | 0 | 80% |
| Testing | 0 | 0 | 100% |
| Evaluation and Future Work | 60% | 33% | 33% |
| User and Deployment Manuals | 20% | 0 | 80% |
| Legal Issues | 100% | 0 | 0% |
| Blog and Monthly Video | 33% | 33% | 33% |
| Overall contribution | 30% | 34% | 36% |

# Appendices

- User manual
  - Teach users to use your application
  - It would be very helpful for the readers if some screenshots are provided to support your explanation.
  - If you need to deliver a website, please remember to provide the URL of the live website, the username, and password to login your website if authentication is required.
  - If there are more than one type of user (e.g. admin and normal user), please provide one account for each type of user.

- Deployment manual (very important)
  - This should provide a step by step guide to show your project partner how to deploy your project after they get your source code.
  - If needed for some difficult parts, please provide some screenshots.

- Legal issues and processes
  - Notes have been uploaded already to Moodle for you make use of in your project.

- Development Blog

- Monthly Video

# Legal Issues and Processes

- The notes for this section will be available in “Legal Implications - Data, IP and GDPR” section on Moodle.
- There is an extensive deck of notes for you to read through.
- Your website should include GDPR and privacy of data in your project; **please document this on your project website under Appendices**.
- Include a statement on the source code license (check with your Project Partners first) and GDPR of data governance.

# Development Blog

- This is an open and public summary of your team's project. It should include your team websites' key assets, progress and experimental design. You can use WordPress, your own preferred blog, or any of the UCL blogging tools.

- The Project Partners have asked to keep track of your progress through the two terms. Ideas for the blog would be to include Interesting research, examples, reference materials, sketches (including from your HCI component), screenshots, algorithm code snippets especially what you have found and customised for the client's solution, and of course your own diagrams and models should be posted here.

- Check it with your Project Partners. You can post your team's development blog on social media with their emailed permission. Each student in a team should be able to contribute to this.

- As they are a diary, you should be aiming to update these blog entries every two weeks (do not submit it all at the end!).

# Monthly Video

- Each month you should ideally be preparing a 1-2 minutes video update about interesting findings and progress made on your experiments.
- Use the PowerPoint PPTX template found in the “Project website, videos and blog” section as the start for your video (3-5 seconds).
- Please remember to set the OneDrive sharing setting of your video file to be ‘Anyone with the link’. If you need a date for your diaries to remember, do this by the 15th of each month.
- Please submit the onedrive link on the form provided on moodle and on the project website. UCL will put approved videos onto their YouTube channels which you can make use of.

# Report Website Marking Criteria

- **Home:** Is the abstract explained well? How is the Gantt chart?
- **Requirements**: Is the project background and goals explained well? How well the project requirements are captured and described? Are personals and use cases described well? Are the MoSCoW requirements clear and specific?
- **Research**: How well the related project reviewed? How well you compared devices, tools, software, API, libraries, algorithms? Are references in place?
- **UI Design** (if applicable): Is the UI design principals, sketches, prototypes documented well?
- **Algorithms** (if applicable)**:** Is the model, experiments, results and discussion described well? Did you build up a quantified scheme to evaluate the performance?
- **System Design**: Has the system architecture design been explained well? Quality of the ER diagram and suitable diagrams.

# Report Website Marking Criteria Continue

- **Implementation**: How well is the implementation of key features described?
- **Testing**: Is there a good testing strategy and a thorough testing?
- **Evaluation**: Is there a good evaluation of the end results of the work? Are the criteria relevant, and the conclusions justified? Is the future plan well documented? Is the plan concrete?
- **Appendix**: Are the user and deployment manuals clear? Are legal issues, development blog, and month videos included?
- **Format**: Does the website have a good format?
- **Clarity**: Is the content well written and readable? How are the spelling and grammar of the report? Does it communicate effectively?

# Mark Range for Report Website

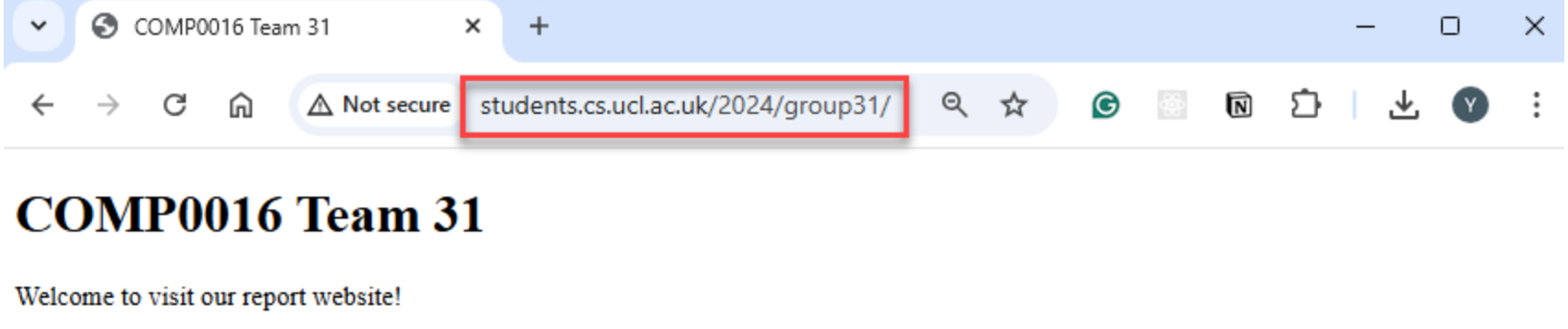
| Mark Range | Description | Criteria |
|---|---|---|
| 90 - 100 | Exceptional | Publishable quality. Close to faultless in documentation. |
| 80 - 89 | Outstanding | Could lead on to publishable work |
| 70 - 79 | Distinction | Very well written with a clear logical structure |
| 60 - 69 | Good (merit) | Clear project write-up with logical structure |
| 50 - 59 | Satisfactory (pass) | Adequate project write-up, lacking clarity or detail in places, or containing irrelevant material |
| 40 - 49 | Weak (BSc and MEng Pass) | Write-up is somewhat incoherent, rushed, contains important omissions, or irrelevant material |
| 30-39 | Inadequate(Fail) | Documentation is poor, unstructured, some parts missing. |
| 0 - 29 | Unacceptable fail | Documentation is substantially absent, badly written, incomprehensible or wrong. |

# Example Websites

- https://students.cs.ucl.ac.uk/2023/group1
- https://students.cs.ucl.ac.uk/2023/group23
- https://students.cs.ucl.ac.uk/2020/group8/
- https://students.cs.ucl.ac.uk/2019/group4/
- https://students.cs.ucl.ac.uk/2018/group24

# Hosting the Project Website

- Hosting Space
  - Each team is assigned a 500 MB space for hosting their COMP0016 report website.
  - Your team's specific directory will follow the format /cs/student/www/2024/group<i>/, where <i> is your team number
  - Others can visit your project website using a URL that follows this pattern: http://students.cs.ucl.ac.uk/2024/group<i>




- Guideline is available in the 'Zoom Room for Online Lectures' section on Moodle.
  - Access Link

# 4. Video

UCL

4

UCL ENGINEERING
Change the world

# Portfolio Video Submission

- 8 minutes
- **mp4 is as the only valid format , 4K or 1080p preferred where possible.**
- If you or your industry partner want to publish your videos on Youtube or other platforms for future reference and the publicity of your project, you must get an email (written clearance) from your Project Partners for permission. If the partner refuse to agree to publish your video, please accept this as their wish for their own reasons.
- Note the video is intended to be a technical description of the project used for assessing the project results.

# Video Content

- The 8 minutes video should include
  - Start with an introduction slide of your project title, team number, team members, and the Project Partners.
  - Introduce the background, problem, and goals of your project
  - List the **key** requirements, please do not list all the requirements
  - Use a system architecture diagram to introduce your solution
  - Describe the main technologies that you use to implement the project
  - Go through the finished functionalities
  - Summarise the achievements

# Notes For Videos

- Please do not submit a silent video. Please have a team member to explain things.
- Please make sure the voice of narrator is clear and do not contain any microphone or background noise.
- Please do not include any background music.
- Please make sure the resolution of the video is high enough to read the text on your slides.

# Video Editing

- Links to editing tools here: https://wiki.ucl.ac.uk/display/UCLELearning/Video+editing+tools
- Apple users - Apple users can make use of iMovie
- Video Editors for PC
  - Windows 10 and Windows 11 users (with latest updates) can make use of Microsoft Video Editor
  - Free VideoDub: http://www.dvdvideosoft.com/products/dvd/Free-Video-Dub.htm#.VY1Nv2DtImR - WATCH FOR ADDITIONAL SOFTWARE INSTALLATION
  - MPEGStreamclip: http://www.squared5.com/svideo/mpeg-streamclip-win.html
  - Lightworks – free: http://www.lwks.com
  - Avidmux: http://fixounet.free.fr/avidemux/index.html - converts video
  - VideoPad Masters: http://download.cnet.com/VideoPad-Masters-Edition/3000-13631_4-10906278.html
  - WeVideo: https://www.wevideo.com