COMP0016: Systems Engineering

Deliverables and Assessment

Dr. Yun Fu
Dr. Dean Mohamedally
Dr. Graham Roberts

UCL Department of Computer Science

UCL ENGINEERING
Change the world

Outline

1. Deliverables of Project Portfolio and Assessment Components
2. Code
3. Website
4. Videos
5. Individual Reports (x2)

Project Portfolio

Portfolio Entry 1: Requirements Gathering, Report Website with Development Blog (weekly entries) and Teams Planning cycle

Portfolio Entry 2: DevOps and High Fidelity Wireframes (online interactive wireframe)

Portfolio Entry 3: Prototype 1 (checking in the last week of term 1) + individual report 1 (submit on March 24)

Portfolio Entry 4: Statements on report website on Legal Implications, Manufacturing Processes and Sustainability (a part of the report website)

Portfolio Entry 5: Code Walkthrough with your Project Partners

Portfolio Entry 6: CS News facing publicity article (submit by an online form)

COMP0016 Assessment Components Weighting

4 Assessment Components	Category	Mark	1st Examiner
Group Coursework (HCI)	Group	5%	Dr Chris Evans
Group Presentation (Elevator Pitch Video Livestream)	Group	5%	All Examiners
Group Portfolio	Group	60% (code 35%, website 20%, video 5%)	Dr Yun Fu
Individual Portfolio	Individual	30%	All Examiners

Deliverables of Final Submission

- Group Portfolio
  - A zip file including three folders (code, website, and video) to **both your Project Partners and to Moodle**.
  - Note: the video folder includes three project demonstration MP4 files (30 seconds, 2 minutes, 8 minutes) and the PowerPoint presentation slides of the 8 minutes video.
  - If your zip file is over 160M, please upload it to **Microsoft OneDrive**, save the link in a text file and submit the text file on Moodle. Please remember to set the OneDrive sharing setting of your zip file to be 'Anyone with the link'.

- Individual Portfolio
  - A zip file including Individual report 1 and 2 **only to Moodle**. Each individual report should include your individual contribution, what difficulties you faced and how you overcame them, and evaluation of your team members and yourself.

Submission Deadline for final submission on Moodle

Final Submission Components	Deliverables	Submission Date
Group Portfolio	Source Code	16:00 Wednesday 24 March 2021
	Report Website	
	Video	
Individual Portfolio	Individual Report 1	
	Individual Report 2	

Marking Procedure

Your team TA is recording progress and achievement each week. It is important that you show your *improvements* in your progress.

- Marking your final submissions
  - Step 1: your team TA and Dr. Yun Fu will arrange a remote meeting to see the demo of your project at the end of term 2.
  - Step 2: The COMP0016 teaching team will read your report website, run your program, watch your videos, read your individual reports. TAs will do the first round of marking.
  - Step 3: Dr. Yun Fu will review the marks and comments recorded by the TAs to do the second round of marking.
  - Step 4: Dr. Dean Mohamedally and Dr. Graham Roberts will do a final consistency check on the marking.

2. Source Code

2

UCL ENGINEERING
Change the world

Source Code Submission

- The code folder should include the following content
  - A readme document to describe how to deploy your code. This readme document can be in the format of word or pdf.
  - Database (if applicable). Export your database as a file and save it in the source code folder
  - The entire source code and build files (any open source external libraries)
- Please frequently update your Github repository that Dr. Yun Fu created for you. If there is a dispute about the code contribution, the commits to the repository might be used as a marking evidence.

Code Assessment

- **Your code will be assessed on**
  - Challenge of the required functionalities
  - Completion state of the required functionalities
  - Quality of your solution
  - User interface & user experience (if applicable)
  - Bug
  - Feedback from the project partners

ID	Requirements	Priority	State
1		Must	✓
2		Must	✓
3		Should	X
4		Could	X
Key Functionalities (must have and should have)		85% completed	
Optional Functionalities (could have)		65% completed	

Mark Range 90-100

- Description
  - Close to Perfect (Exceptional 1st)

- Criteria
  - A significant contribution to the field
  - An original and/or a model solution to the problem in question
  - Challenging goals and all the required functionalities are delivered
  - Close to faultless in execution
  - Results released, or ready to be released as a paper or product.

- Comments
  - This represents a really outstanding achievement.
  - The project needs to clearly stand out above others.
  - A mark in this range is hard to achieve, not impossible but certainly **rare**.
  - It needs a highly ranked short paper, journal paper or conference paper with your client and supervisors to be accepted for publication by the marking date (late April).

Mark Range 80-89

- Description
  - Outstanding (Excellent 1st)

- Criteria
  - A useful contribution to the area
  - Challenging goals and all the required functionalities are delivered
  - Only minor faults in execution
  - Results close to being releasable as a product or high-quality working prototype

- Comments
  - This represents a project that stands out as excellent in most respects but doesn’t fully meet the criteria for the top range.
  - We would expect a small number of projects (5-10% maybe) to be in this range.

Mark Range 70-79

- Description
  - Very Good (1st)

- Criteria
  - A good outcome that has found and built a feasible solution to the problem posed.
  - Challenging goals and nearly all the required functionalities are delivered
  - Some small faults in execution or understanding, but largely correct.
  - Capable of being released as a product with some additional work.

- Comments
  - This represents a first class project that means most of things have been done well, but there are some faults or criticisms.
  - We would expect a number of projects (10%-20% maybe) to achieve this level.

Mark Range 60-69

- Description
  - Good (2:1)

- Criteria
  - A solid set of results and the main problem largely solved.
  - Challenging goals and all the must have and should have features are delivered
  - There are some ambiguities or faults
  - The results are able to show a feasible prototype but there are typically some limitations or omissions.

- Comments
  - A good result, that is well on the way to delivering a complete working version of the system, but is not fully complete or finished.
  - We would expect the majority of projects to be at this level.

Mark Range 50-59

- Description
  - Satisfactory (2:2)

- Criteria
  - A partial solution that addresses most of the key issues but is not complete
  - All the must-have features are delivered
  - Project execution not particularly ambitious, or not entirely completed
  - The results are good enough, and the basic features working, but a fair amount still to do.

- Comments
  - A satisfactory but limited result. The core features are in place but may be buggy and not that well defined.
  - We would expect a minority number of projects to be in this range (10-20% maybe).

Mark Range 40-49

- Description
  - Weak (BSc Pass, MEng Fail)

- Criteria
  - A basic solution that shows some progress but is some way from completion.
  - Some must-have features are implemented.
  - The results show that at least a solution is possible but there are significant omissions and flaws.

- Comments
  - A just about adequate project, in that it has achieved enough to get a BSc pass mark, but well below expectations.
  - We would expect no more than 2 teams in this range.

Mark Range 30-39

- Description
  - Inadequate (Fail)

- Criteria
  - No working or partially working solution.
  - Concrete achievements very few, project goals not nearly achieved
  - Project has been run badly

- Comments
  - You have failed. Almost certainly due to lack of effort more than anything else.
  - We do not expect to have to fail any projects but will do if the results require it.

Mark Range 0-29

- Description
  - Very poor (Bad fail)

- Criteria
  - No solution has been identified and the group is not capable of progressing.
  - All aspects of the project have been handled badly.
  - Actual achievements may be very few.
  - Project is dysfunctional.

- Comments
  - Inexcusable result, that should never happen.
  - We don’t want any projects in this range and the monitoring/supervision process should prevent it happening.

3. Report Website

3

UCL ENGINEERING
Change the world

Website Structure

- Home
- Requirements
- Research
- Algorithms (if applicable)
- UI Design (if applicable)
- System Design
- Implementation
- Testing
- Evaluation
- Appendices
- Development Blog – External Link on a public site (**this must be updated weekly**)

Website Submission

- The website folder should include all the HTML, CSS, JavaScript files.

- 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

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

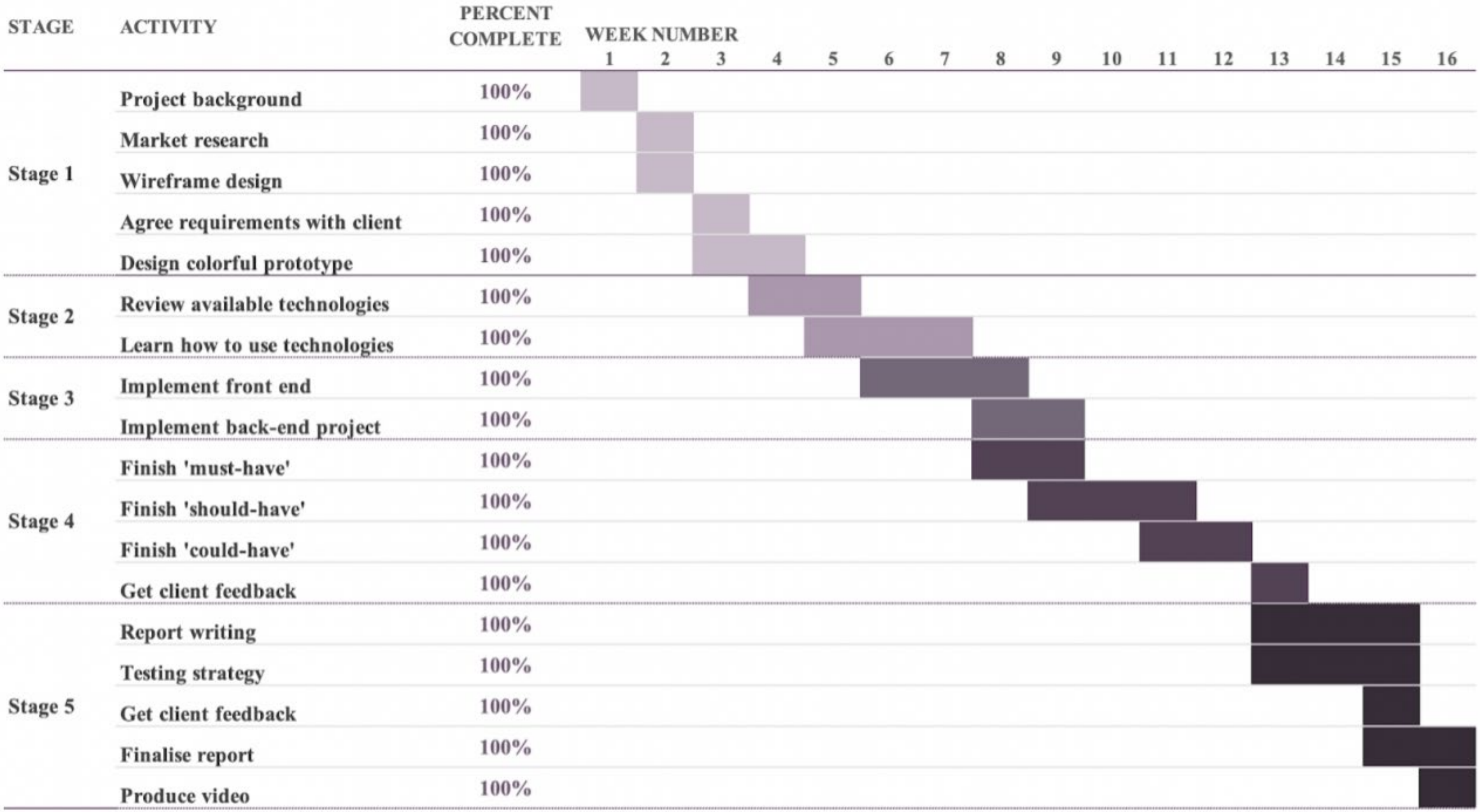
- The development team
  - (including the team photos where possible) name, email, role or main contribution of each member. The roles include client liaison, UI design, researcher, programmer, report editor, tester.
  - You are welcome to include interesting background about yourself, as well as your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

- Project management
  - Gantt chart (from October 5 2020 to March 24 2021)

Gantt Chart Example 1

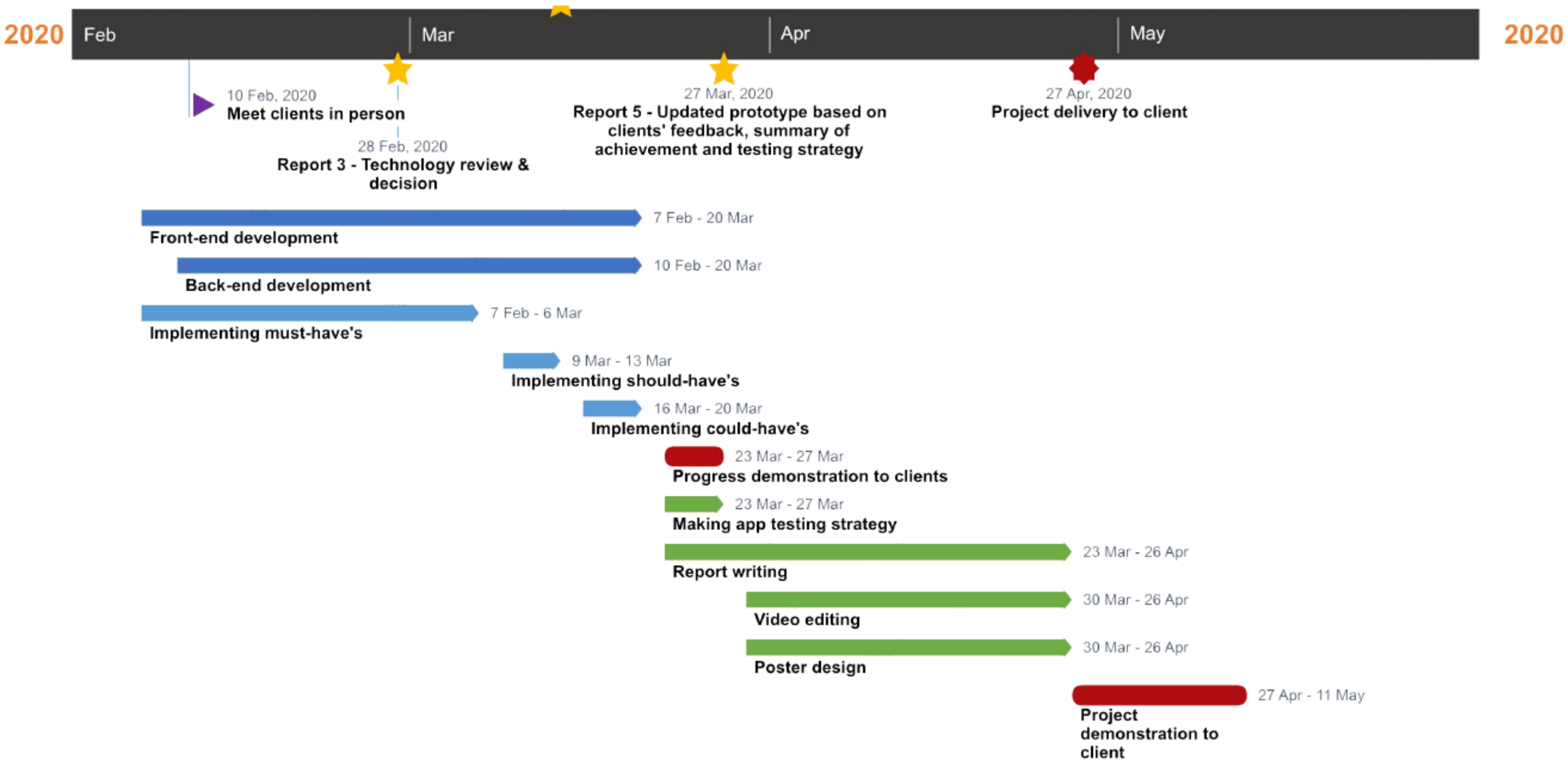
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13+
Requirement Analysis	■												
Sketching and Prototyping		■											
HCI Design		■	■										
Research Technologies				■									
Implement Front-End					■	■	■						
Implement Back-End						■	■	■					
Testing									■				
Bug Fixing										■			
Refining											■		
Writing Software Docs												■	
Writing Project Report													■
Create Video and Poster													■

Gantt Chart Example 2



STAGE	ACTIVITY	PERCENT COMPLETE	WEEK NUMBER															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Stage 1	Project background	100%	■															
	Market research	100%		■														
	Wireframe design	100%		■														
	Agree requirements with client	100%			■													
	Design colorful prototype	100%			■	■												
Stage 2	Review available technologies	100%				■	■											
	Learn how to use technologies	100%					■	■	■									
Stage 3	Implement front end	100%						■	■	■								
	Implement back-end project	100%								■	■							
Stage 4	Finish 'must-have'	100%								■	■							
	Finish 'should-have'	100%									■	■	■					
	Finish 'could-have'	100%											■	■				
	Get client feedback	100%													■			
Stage 5	Report writing	100%													■	■	■	
	Testing strategy	100%													■	■	■	
	Get client feedback	100%															■	
	Finalise report	100%															■	■
	Produce video	100%																■

Gantt Chart Example 3



Requirements

- Project background and Client introduction

- Project goals

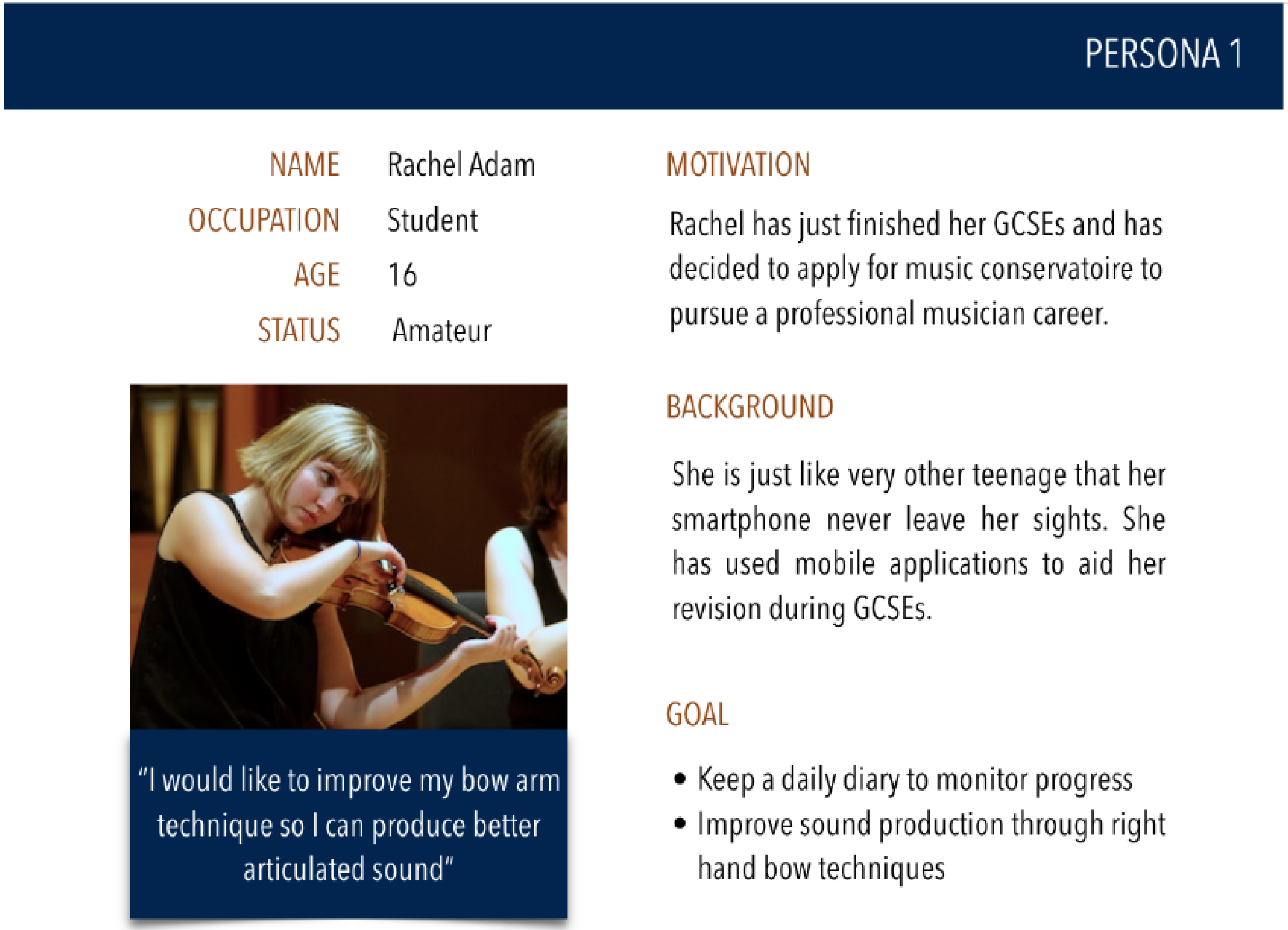
- Requirement gathering
  - How did you collect the requirements?
  - Did you design any survey? How did you analyse the survey data?

- Personas
  - Typical users of your project

- Use cases
  - 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.

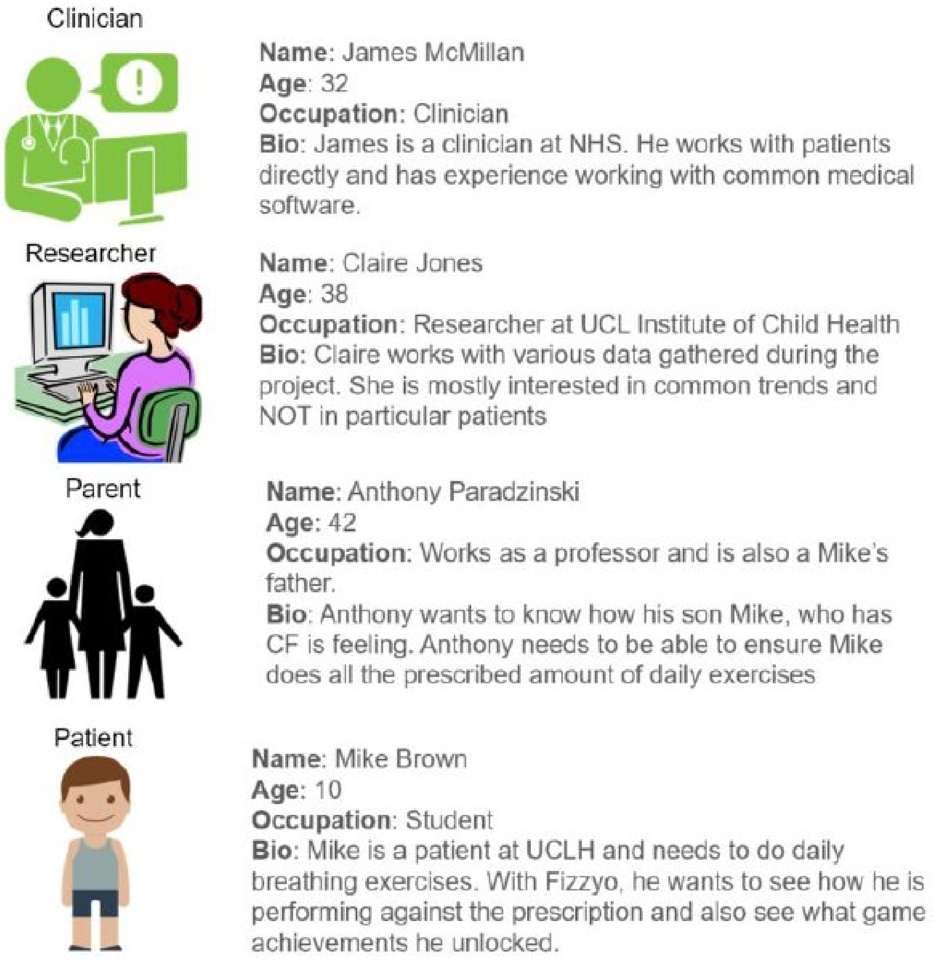
BACKGROUND

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

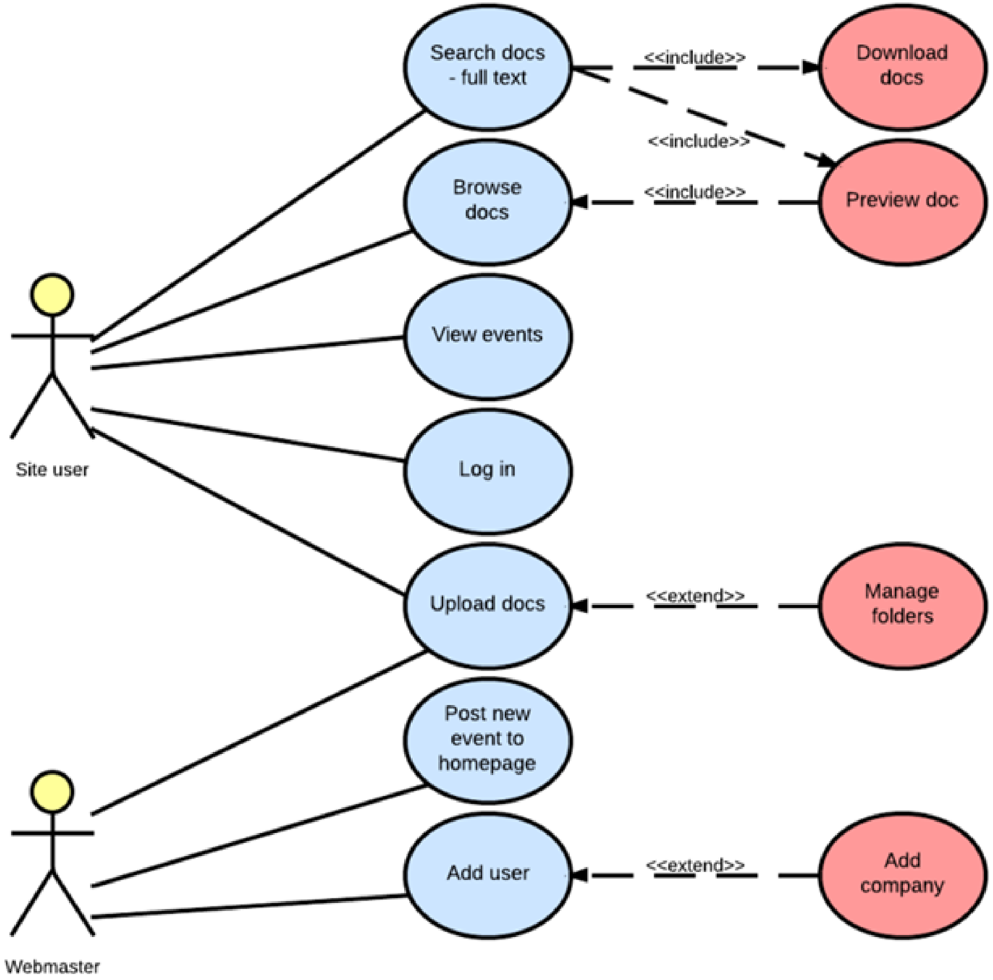
Parent

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

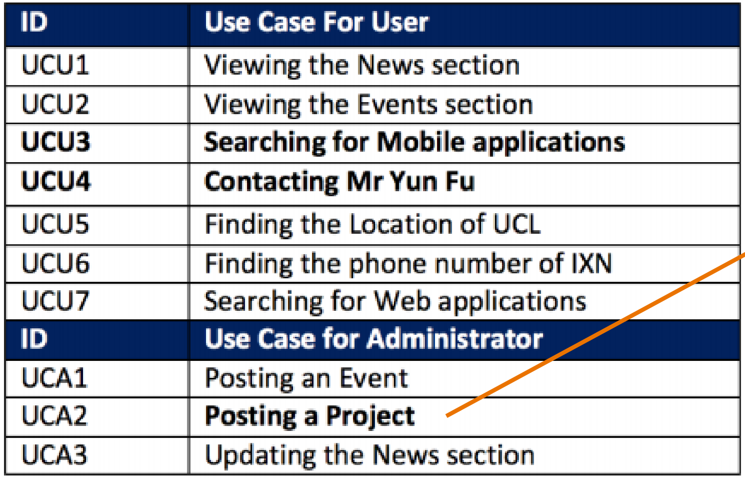
Name: Mike Brown
Age: 10
Occupation: Student
Bio: Mike is a patient at UCLH and needs to do daily breathing exercises. With Fizzyo, he wants to see how he is performing against the prescription and also see what game achievements he unlocked.

Use Case Diagram Example



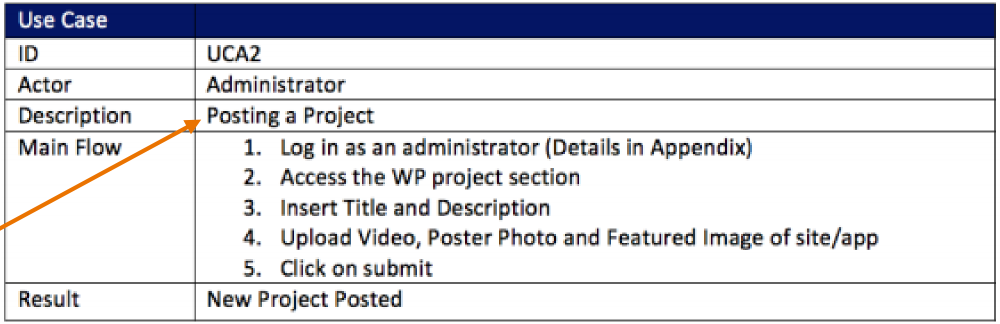
Use Case List

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

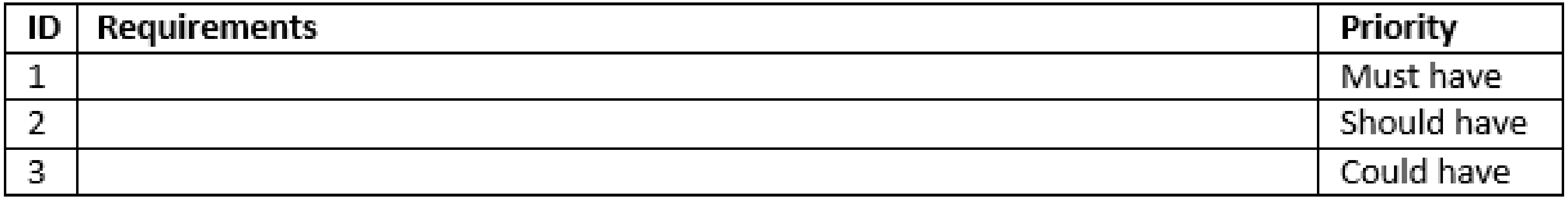
Use case Description



Use Case	
ID	UCA2
Actor	Administrator
Description	Posting a Project
Main Flow	1. Log in as an administrator (Details in Appendix) 2. Access the WP project section 3. Insert Title and Description 4. Upload Video, Poster Photo and Featured Image of site/app 5. Click on submit
Result	New Project Posted

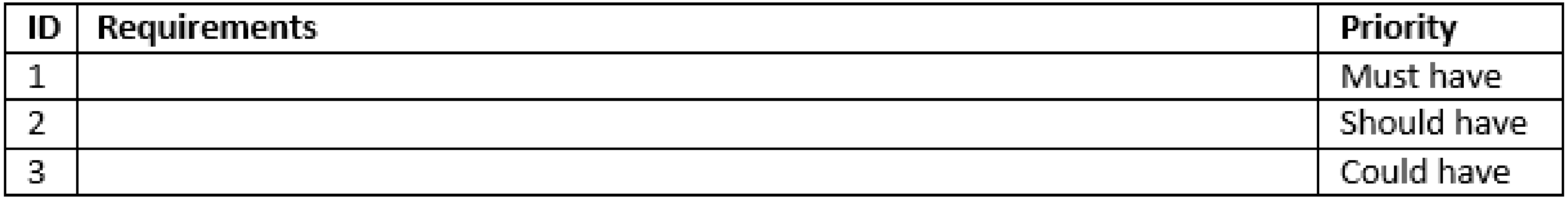
MoSCoW requirement list

- **Functional** requirements



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

- **Non-functional** requirements



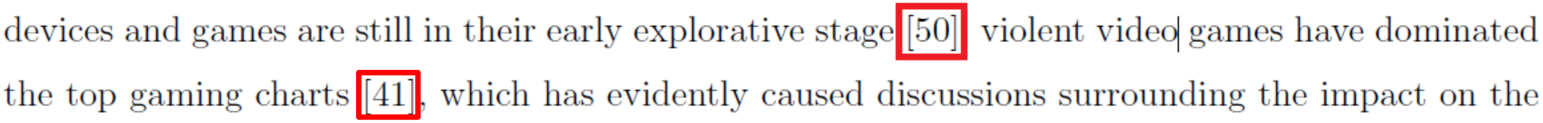
ID	Requirements	Priority
1		Must have
2		Should have
3		Could have

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



devices and games are still in their early explorative stage [50] violent video games have dominated the top gaming charts [41], which has evidently caused discussions surrounding the impact on the

  - 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
  - Use plots or tables to show the performance comparison results

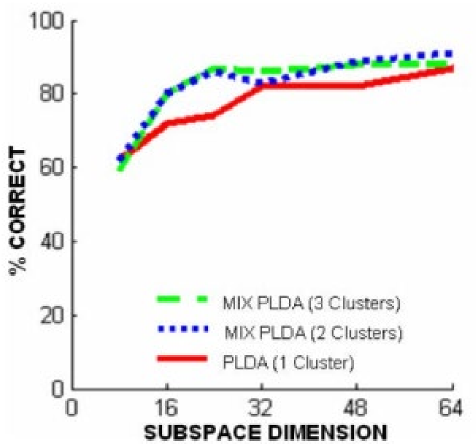
- Discussions
  - Why the algorithm fails for some test examples
  - Suggestions to improve the performance

- Conclusion

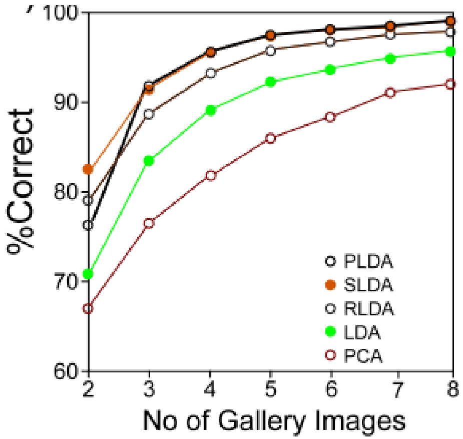
- References

Plot and Table Example

Investigation of the optional Hyperparameters Plot Example 1



Investigation of the optional Hyperparameters Plot Example 2



ROC Curve Example

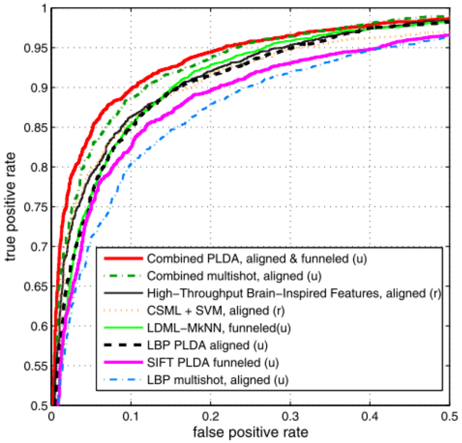
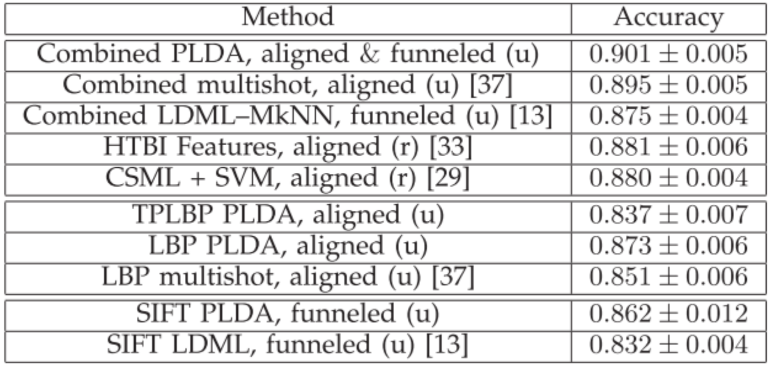


Table Example

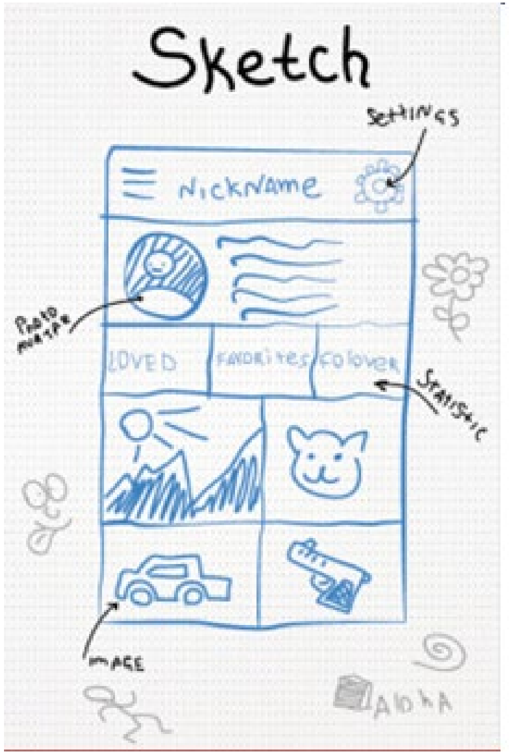


Method	Accuracy
Combined PLDA, aligned & funneled (u)	0.901 ± 0.005
Combined multishot, aligned (u) [37]	0.895 ± 0.005
Combined LDML–MkNN, funneled (u) [13]	0.875 ± 0.004
HTBI Features, aligned (r) [33]	0.881 ± 0.006
CSML + SVM, aligned (r) [29]	0.880 ± 0.004
TPLBP PLDA, aligned (u)	0.837 ± 0.007
LBP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

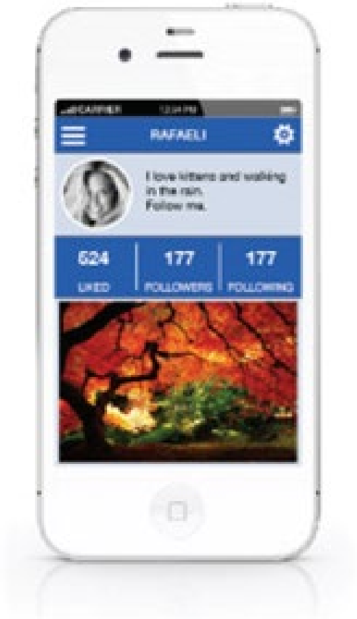
UCL ENGINEERING
Change the world

User Interface Design (if applicable)

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



Wireframe

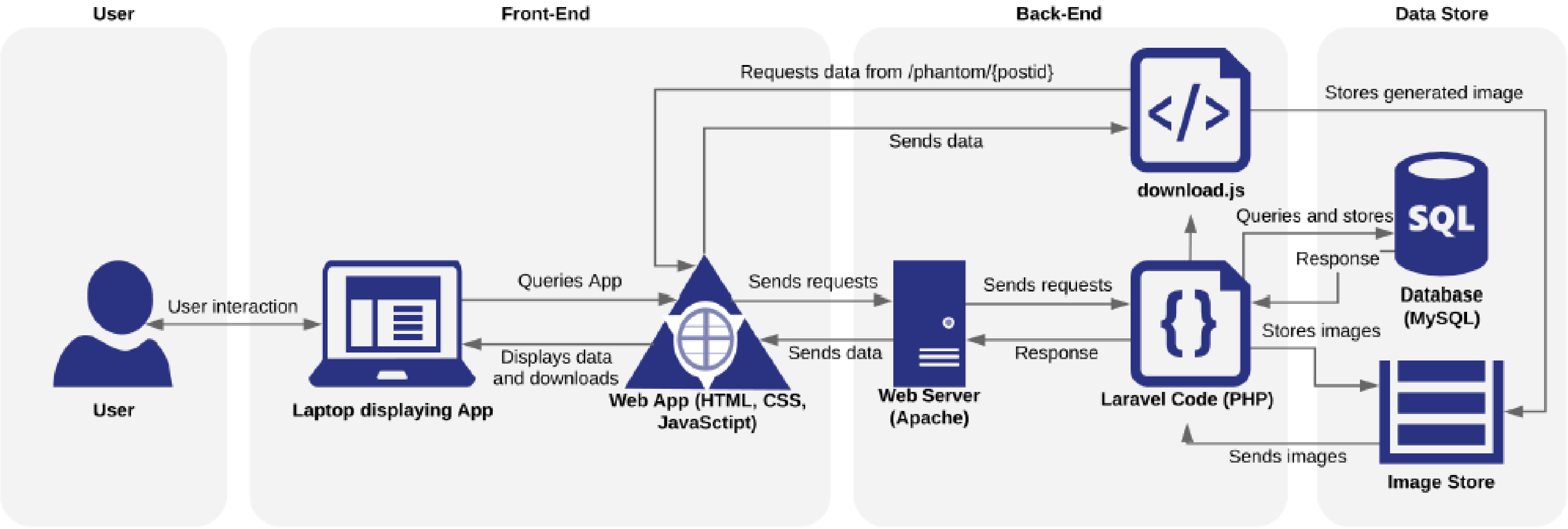


I love kittens and walking in the rain. Follow me.

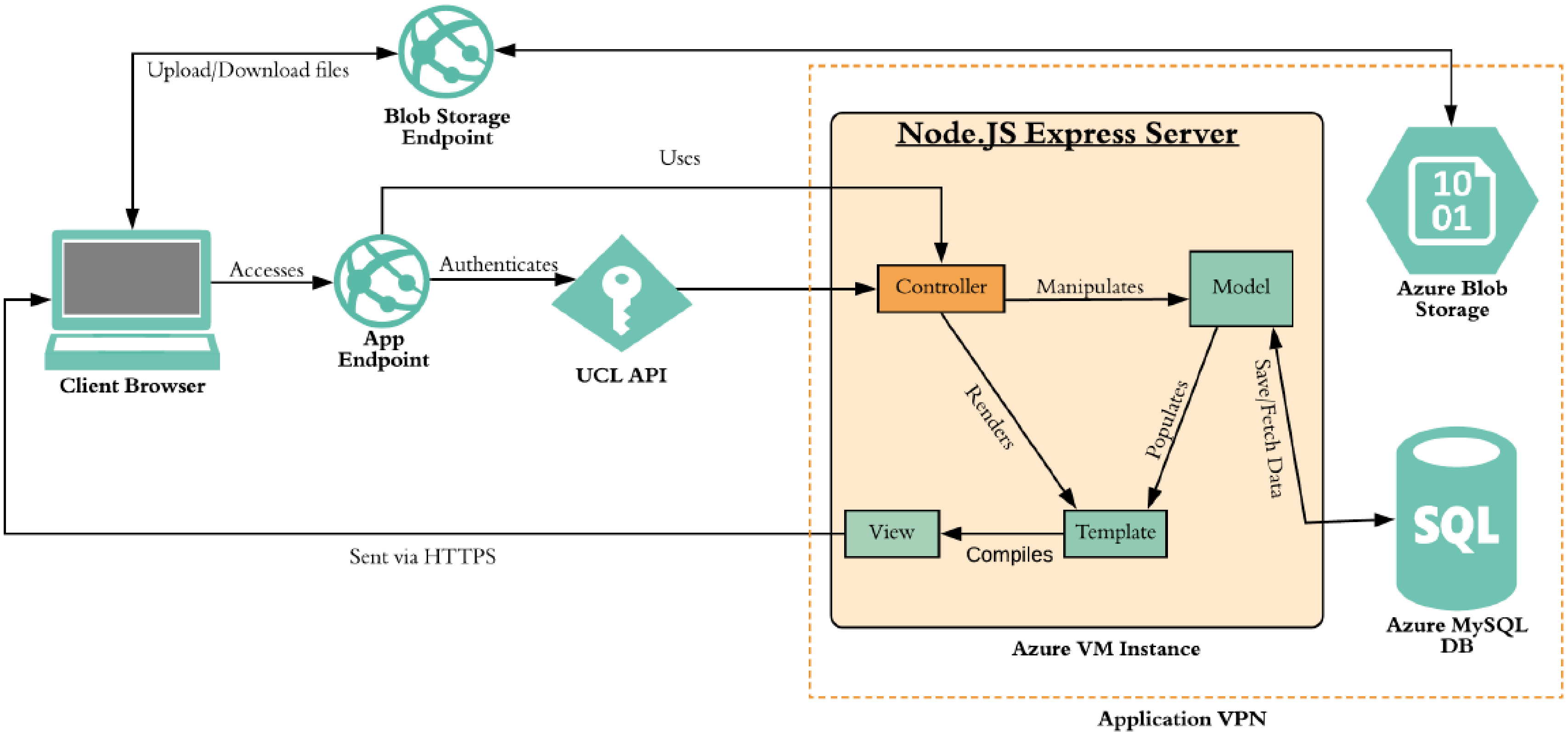
System Design

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

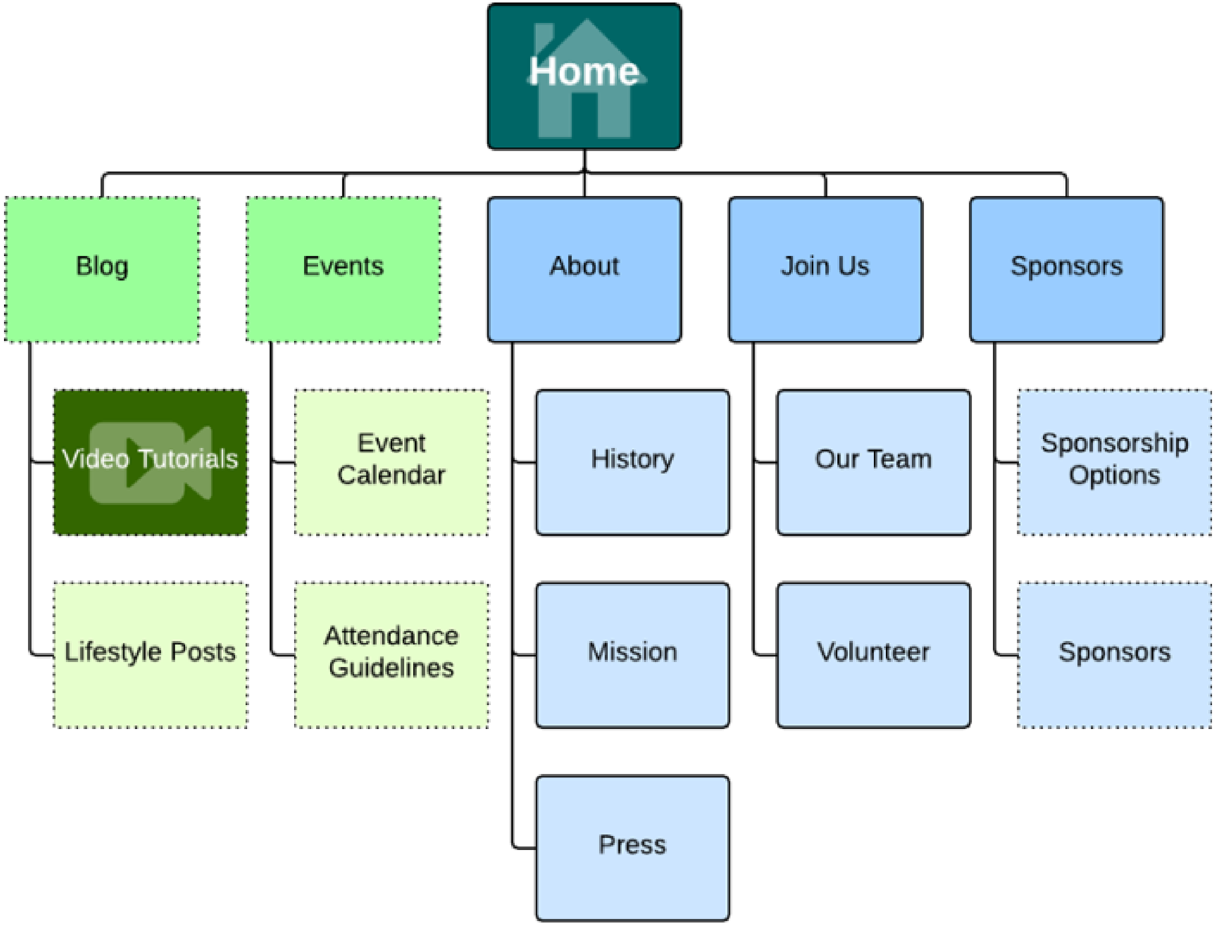
System Architecture Example 1



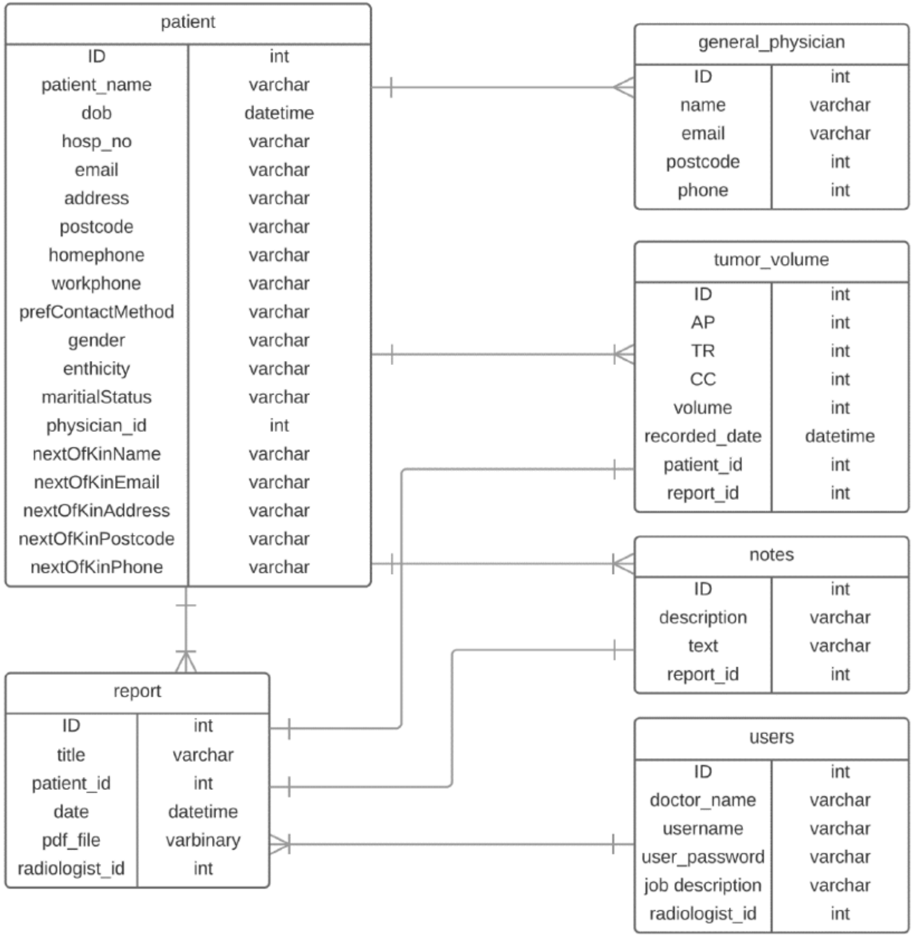
System Architecture Example 2



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 export, 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
  - Testers
  - Test cases
  - Feedback from testers and project partners

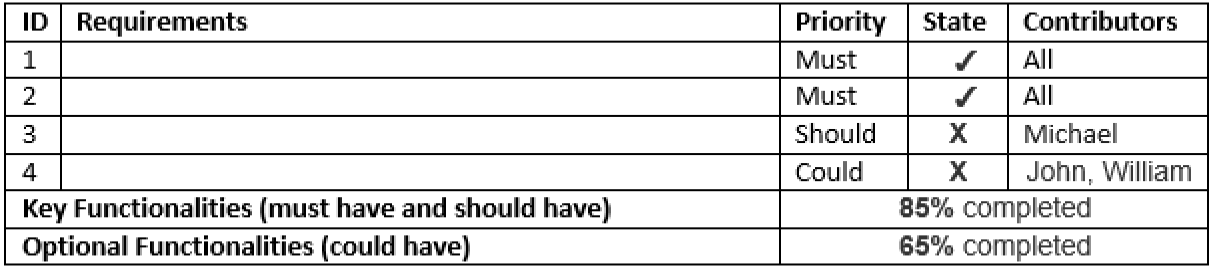
- For each test, you should explain why you do this test, which test tool you use, how you conduct the test, what results you get, and your analysis or conclusion to the results.

Evaluation

- Summary of achievements
  - An achievement table to list the MoSCow features, the completed states, and contributors
  - A list of known bugs
  - 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

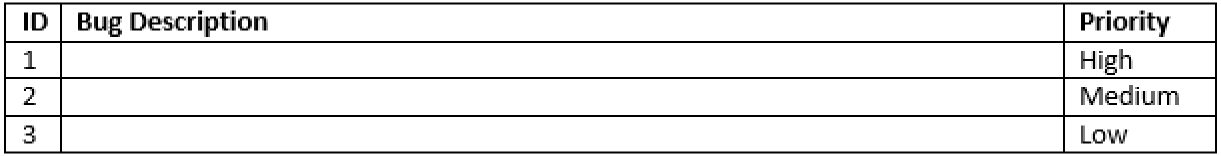
Example of Achievement Table and Known Bug List

- Achievement table



ID	Requirements	Priority	State	Contributors
1		Must	✓	All
2		Must	✓	All
3		Should	X	Michael
4		Could	X	John, William
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 Distribution Table Example

Work packages	John	Michael	William
Project Partners liaison	33%	33%	34%
Requirement analysis	34%	33%	33%
Research and Experiments	50%	0	50%
UI Design (if applicable)	20%	10%	70%
Coding	20%	0	80%
Testing	0	0	100%
Report Website	60%	33%	33%
Video Editing	50%	0	50%
Overall contribution	30%	34%	36%
Main Roles (maximum three for each member)	Researcher, Report Editor, Front End Developer	Back End Developer, Report Editor	UI Designer, Front End Developer, Tester

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 you need to deliver a mobile app (Android or cross platform app), please provide a link to download the apk file. Please also provide username and password 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 industry 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
  - Dr. Dean Mohamedally will give a lecture in term 2 to cover this part.

Legal Issues and Processes

- This section will be defined by Dr Dean Mohamedally
- It includes legal statement on the source code licence, IP rights, GDPR of data governance.
- It also includes manufacture processes and the deployment costing for sustainability (especially if it is on the cloud).
- Notes of these topics will be given on Moodle in Term 2.

Development Blog

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

- The clients 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.

- You can post your team's development blog on social media as well as share with your clients, with your clients permission. Each student in a team should be able to contribute to this.

- Dr Mohamedally will be collecting these development blogs. As they are a diary, you should be submitting these blog entries weekly. Please set these up now to use.

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**: Is the UI design principals, sketches, wireframes documented well?
- **System Design**: Has the system architecture design been explained well? Quality of the ER diagram and suitable diagrams.
- **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?
- **Future Work**: Is the future plan well documented? Is the plan concrete?
- **Appendix**: Are the user and deployment manuals clear?
- **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 Pass, MEng fail)	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.

4. Video

4

UCL ENGINEERING
Change the world

Video Submission

- The video folder should include three videos (30 seconds, 2 minutes, 8 minutes) and the PowerPoint presentation slides of the 8 minutes video

- **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 and send the email (written consent) to Dr Dean Mohamedally. If the client refuse to agree to publish your video, please accept this as their wish for their own reasons.

Video Content

- 30 seconds video only need to demonstrate the most important functionalities.

- 2 minutes video should include
  - Start with an introduction slide of your project title, team number, team members, and the Project Partners.
  - Introduce **briefly** the background, problem, and goals of your project.
  - Use a system architecture diagram to describe your solution **briefly**
  - Demonstrate the key functionalities

- 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 narrator to explain things.

- Please make sure the voice of narrator is clear and do not contain any microphone or background noise.

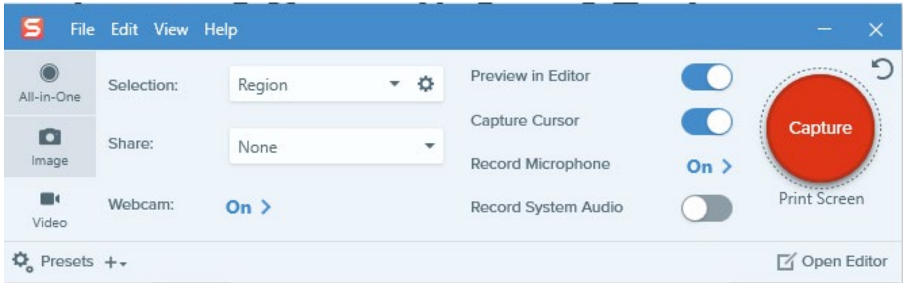
- Please do not include any background music.

- Please make sure the resolution of the video is high enough to read the text on your slides.

Video Editing

- Apple iMovie

- Snagit
  - Free to trail for 15 days
  - Click this link to download Snagit
  - Features: record voice, switch between webcam and screen recording, video editing
  - Tutorial: How to Document a Process With Video



Selection: Region

Share: None

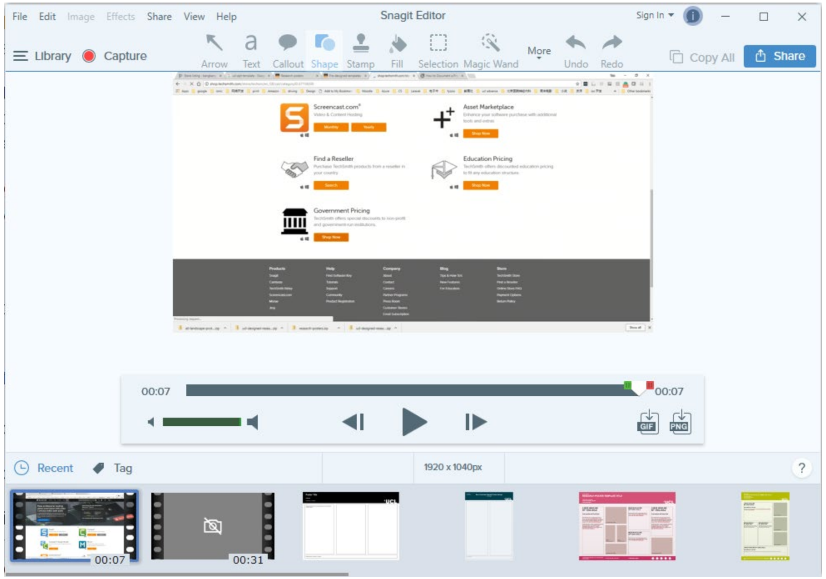
Webcam: On

Preview in Editor ●

Capture Cursor ●

Record Microphone On

Record System Audio ○



5. Individual Report and Contribution

5

UCL ENGINEERING
Change the world

Individual Reports 1 and 2

- Two individual reports should be uploaded to Moodle before 16:00 Wednesday 24 March, **each is 2-5 pages**

- **pdf is the only valid format**

- Individual report 1 should cover the last five weeks of term 1.

- Individual report 2 should cover the whole development (last five weeks of term 1, Christmas holiday, and the term 2)

- Please use section numbers such as 1 and 1.1 to organise your report

Content of Individual Reports 1 and 2

- The project title, team number, your name, and your student ID
- List your main personal contributions to the project (e.g. Project Partner liaison, research, UI design, coding, testing, website, video editing)
- The main difficulties you faced and how you overcame them
- Assessment of each team member (including yourself)
  - List the strengths and weaknesses (e.g. reliability, technical skills, communication skills, document writing etc.) of each member (**including yourself**) and the role that each member is best suited to (Project Partner liaison, UI designer, researcher, programmer, report editor, tester)
  - Please evaluate the performance of your team members and yourself via a number between 1 (poor) and 10 (excellent). Please give some evidence to explain why you give the number.
  - (**optional**) If you have a different opinion about the individual contribution table on the report website, please indicate which part you do not agree with and explain why.

Individual Assessment

- **Your individual performance will be assessed on**
  - The individua contribution distribution table on your report website
  - Your individual report
  - Assessment of you in your team members’ individual reports
  - TA’s feedback

Variations in Marks for Group Components

- By default all team members have the same team marks for the team work (code, website, video).

- **However, this is not fixed and marks can be reduced for individuals.**
  - This occurs if a team member did not participate, or failed to properly contribute such as by leaving tasks undone. This leads to reduced team marks, **potentially down to zero if no contribution was made**.

- In practice we have found that in the large majority of teams all individuals contribute relatively evenly, so not many variations are needed.

Variations in Final Marks for Team Members

- Members of the same team do not get the same mark due to the 30% individual portfolio mark.

- The individual portfolio mark is based on the weighted sum of the team work (code 35%, website 20%, video 5%) and then is adjusted based on your individual contribution to the project.