

Coursework Specification

The coursework is worth 70% of the total module mark. Delivery of the coursework is divided into four three-week Sprints. You will also be evaluated for your individual contribution to the project. The marks for the coursework are divided as follows:

Title	Deadline	Marks
Sprint 1	4 February	20%
Sprint 2	25 February	20%
Sprint 3	18 March	20%
Sprint 4	22 April	20%
Individual Reflection	22 April	20%

In this coursework you are tasked with creating an application for the smooth and efficient operation of a business.

Your application must be built using the tools and techniques taught in the module, although there is some flexibility.

You will work on the project as a Scrum team. Details on Scrum are provided in your lectures, including an FAQ on how to apply Scrum in the module. You will work as teams. You will therefore need to adopt both design methods (user stories, use cases) and project management methods (Agile, Kanban). You will also need to become familiar with the tools of a typical software development team (Visual Studio, Git, Docker) and some of the difficult human and technical issues around collaborating on and continuous improvement of online applications. Fundamentally, you should aim to develop a [CRUD application](#). A CRUD application is effectively a front-end to a database, providing a method to create, read, update, and delete data from a web site to a database on a server.

- Your web front-end should use PUG
- Your web back- end must run via Node.js
- Your front-end and back-end must communicate via Express.js
- Your application must use a database for storage
- Your application must be version controlled using Git and publicly accessible on GitHub
- Your application must be deployable as a Docker container

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

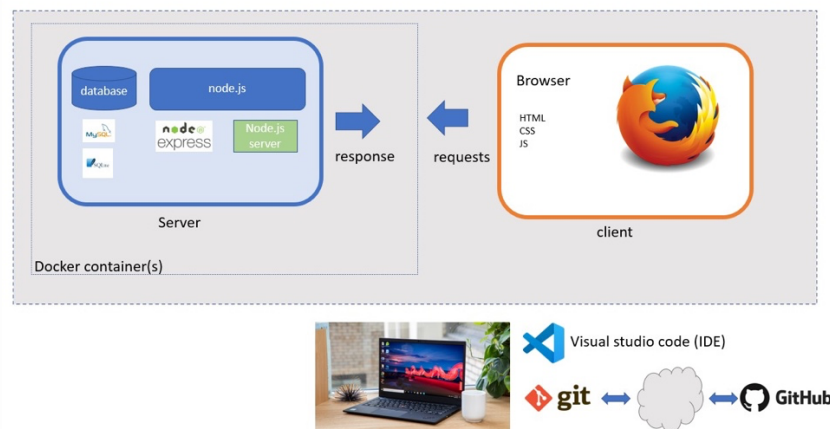


Figure 1 – a simple example

You work for an organisation that requires reporting on population information. You have been tasked with designing and implementing a new system to allow easy access to this population information. The organisation has provided you with an SQL database to work from available:

<https://downloads.mysql.com/docs/world-db.zip>

You need to do the following for your project:

- Identifying the requirements (functional and non-functional)
- Prioritising the requirements (if applicable)
- Task allocation
- Identifying the scope of your project
- Identifying the stakeholders
- Risk management

The organisation has asked for the following reports to be generated:

- All the countries in the world organised by largest population to smallest
- All the countries in a continent organised by largest population to smallest
- All the countries in a region organised by largest population to smallest
- The top `N` populated countries in the world where `N` is provided by the user
- The top `N` populated countries in a continent where `N` is provided by the user
- The top `N` populated countries in a region where `N` is provided by the user
- All the cities in the world organised by largest population to smallest
- All the cities in a continent organised by largest population to smallest
- All the cities in a region organised by largest population to smallest
- All the cities in a country organised by largest population to smallest
- All the cities in a district organised by largest population to smallest
- The top `N` populated cities in the world where `N` is provided by the user
- The top `N` populated cities in a continent where `N` is provided by the user
- The top `N` populated cities in a region where `N` is provided by the user

- The top `N` populated cities in a country where `N` is provided by the user
- The top `N` populated cities in a district where `N` is provided by the user
- All the capital cities in the world organised by largest population to smallest
- All the capital cities in a continent organised by largest population to smallest
- All the capital cities in a region organised by largest to smallest
- The top `N` populated capital cities in the world where `N` is provided by the user
- The top `N` populated capital cities in a continent where `N` is provided by the user
- The top `N` populated capital cities in a region where `N` is provided by the user
- The population of people, people living in cities, and people not living in cities in each continent
- The population of people, people living in cities, and people not living in cities in each region
- The population of people, people living in cities, and people not living in cities in each country

Additionally, the following information should be accessible to the organisation:

- The population of the world
- The population of a continent
- The population of a region
- The population of a country
- The population of a district
- The population of a city

Finally, the organisation has asked if it is possible to provide the number of people who speak the following the following languages from greatest number to smallest, including the percentage of the world population:

- Chinese
- English
- Hindi
- Spanish
- Arabic

Country Report

A country report requires the following columns:

- Code
- Name
- Continent
- Region
- Population
- Capital

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

City Report

A city report requires the following columns:

- Name
- Country
- District
- Population

Capital City Report

A capital city report requires the following columns:

- Name
- Country
- Population

Population Report

For the population reports, the following information is requested:

- The name of the continent/region/country
- The total population of the continent/region/country
- The total population of the continent/region/country living in cities (including a %)
- The total population of the continent/region/country not living in cities (including a %)

Additional functionality

- Being able to add new info (dataset) to the database
- Applying security measures/login credentials (having a username and password)
- Extra yet relevant functionality in your application

Group Submission

The coursework **must** be delivered by a group. The aim of the module is to assess your ability to work as a team to deliver a product. Therefore, **the majority of your coursework grade will be based on your team's ability to work together using the methods defined in the module.**

The submission is monitored weekly during lab stand-up meetings, and formally via the code-review assessment points. Your final submission is delivered via your GitHub repository.

Individual Assessment

Individual contributions to the team will be assessed based on attendance at the various meetings and code reviews, and via the metrics gathered from tools such as GitHub. **Individual contributions can lead to a scaling of the overall coursework grade if the module team have evidence that illustrates a lack of contribution to the team deliverable.**

The coursework **must** be delivered as part of a team. **If anyone is dismissed from their team this means they cannot deliver the coursework and will fail.** Dismissal from a team involved the following process:

- An individual is evidenced as breaching the code of conduct as set-out by the student team
- Evidence is presented at the next available meeting with a member of the module delivery team
- The individual evidenced will have the opportunity to evidence mitigating circumstances either to the student team or privately to a member of the module delivery team
- The module delivery team retains the right to the final decision of whether the dismissal is warranted

Any dismissed team member has a week to appeal the decision to the module team with suitable evidence provided.

Code Review Meetings

Each group will undertake **four** graded code reviews:

1. Week 3 (20% of CW mark)
2. Week 6 (20% of CW mark)
3. Week 9 (20% of CW mark)
4. Week 12 (20% of CW mark)

The meetings will take place during the lab sessions. Each group will be given **10 minutes maximum** for the code review. Your group will be **allocated a time for the code review**. The details of the individual review points are below. These meetings **must be attended** at the **stated time**. Guidelines for grading the group:

- **Being late** for the meeting or **not being ready** when the meeting starts will result in the grade for that review being capped at 40%
- **Not attending** the meeting will mean the code review will be marked at 0%
- **All team members** should attend the code review, however commitments and other considerations will be taken into account. **Individuals attendance at reviews will be monitored** to ensure the team is contributing collectively to the project.
- **Being ready** means that you are ready to present the points for the code review. This means that you have a computer with the various tools logged into (e.g., GitHub, Travis CI, etc.) and a building version of the application in IntelliJ.

Deliverables

Sprint 1: Lab of Week 3

The aim of this sprint is to check that the project workflow is set-up for the team. You may choose to meet some of the feature requirements during this review point, but it is not as necessary.

Checklist Submission 1 (16% of CW mark)

The following must be in place:

- GitHub project for coursework set-up
- Product Backlog created
- Dockerfile for project set-up and works
- Correct branches for GitFlow workflow created - includes `master`, `develop`, and `release` branches
- First release created on GitHub
- Code of Conduct defined

Graded Criteria Submission 1 (4% of CW mark)

The following criteria will be assessed for overall quality:

- Metrics from GitHub. Also used to assess individual contribution
- Code quality including comments

Module name: Software Engineering (CMP020N204S)
Lecturer: Dr Sobhan Y. Tehrani
Coursework

Sprint 2: Lab of Week 6

The aim of this sprint is to check that task management is set-up and that the initial requirements gathering has taken place via user stories and use cases. You should have completed at least 25% of the work for the project at this point based on your own estimates.

Checklist Submission 2 (14% of CW mark)

The following must be in place:

- Issues being used on GitHub
- Tasks defined as user stories
- Project integrated with Zube.io
- Kanban/Project Board being used
- Sprint Boards being used
- Full use cases defined
- Use case diagram created

Graded Criteria Submission 2 (6% of CW mark)

The following criteria will be assessed for overall quality:

- Metrics from GitHub. Also used to assess individual contribution
- Code quality including comments
- Correct usage of branches
- Continuous integration working
- Use cases well defined
- Project requirements met

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Sprint 3: Lab of Week 9

The aim of this Sprint is to have a working prototype of your application, showing good progress in the technical deliverables of the project, and providing a proof of concept for your idea.

Checklist Submission 3 (12% of CW mark)

The following must be in place:

- Suitable unit tests defined
- Suitable integration tests defined
- Tests running on Travis CI
- Updated task board of user stories showing progress in the technical deliverables
- Github repository showing work from all members of the team
- Implementation progress towards final version of application:
 - Database design
 - Database implementation with pre-filled data
 - HTML pages available generated from PUG, perhaps with some static content
 - At least some dynamic content being pulled from your database to your templates via Pug - concentrate on views of your data

Graded Criteria Submission 3 (8% of CW mark)

The following criteria will be assessed for overall quality:

- Metrics from GitHub. Also used to assess individual contribution
- Code quality including comments
- Correct usage of branches
- Continuous integration working
- Kanban/Project Board being used
- Quality and coverage of unit tests
- Project requirements met

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Sprint 4: Lab of Week 12

The aim of this Sprint is to deliver the final version of your application. The final version of your application will be pulled automatically from GitHub. The final submission will be assessed based on the following criteria.

Checklist Submission 4 (10% of CW mark)

The following must be in place:

- Overall application quality and success at meeting the aims set out during the design process
- Metrics from GitHub: used to assess individual contribution
- Application deployable using a docker-compose file, data can be provided separately
- Final version of the task board and project requirements met
- Deployment working
- Bug reporting system set-up
- A group report (maximum 10 pages), where you need to present your work as a team

Graded Criteria Submission 4 (10% of CW mark)

The following criteria will be assessed for overall quality:

- Metrics from GitHub. Also used to assess individual contribution
- Code quality including comments
- Correct usage of branches
- Continuous integration working
- Kanban/Project Board being used
- Quality and coverage of unit tests
- Project requirements met

Individual Reflection Report: End of Week 12

Your individual reflection report should contain the following sections:

1. Introduction
 - Short description of the project
2. Design process
 - What design techniques did you use?
 - Which ones did you find the most useful for yourself (a) as an individual and (b) for the group
3. Development
 - What did you personally contribute to the project?
 - How happy were you with your own contribution?
 - Is there anything you would have liked to have done more of?
 - What did you find easy?
 - What did you find difficult?
 - What would you do differently next time?
4. Teamwork
 - How well do you think you worked as a team?
 - What did you learn about teamwork from this project?
 - What percentage of the overall work for the project would you estimate that you personally did?
5. Reflection
 - Lesson learnt
 - What would you like to do more of following this project?

Module name: Software Engineering (CMP020N204S)
Lecturer: Dr Sobhan Y. Tehrani
Coursework

Marking Scheme:

Sprint 1

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
Task definition the completeness and suitability of the initially defined tasks.	Tasks represent clearly the main tasks to be completed and are suitable for the application proposed.	Tasks may either be lacking a complete set of those expected at this stage, or there might be some unsuitability in the tasks defined.	Tasks lack completeness and there are tasks defined that are unsuitable for the application proposed.	No tasks provided, or tasks completely unsuitable and incomplete.
Code of Conduct the suitability of the Code of Conduct and its ability to support the team's work.	Code of Conduct suitably covers the problems that can arise and how the team will resolve issues when they arise.	Code of Conduct covers most issues the team will face and provides suitable guidance of how to resolve these.	Code of Conduct only covers some significant issues and/or doesn't provide clear guidance of how to proceed.	Code of Conduct not provided, or unsuitable for managing the team.
Technical aspects the completeness of technical deliverables.	Both GitHub repository and task board created and all members have joined and contributed.	GitHub repository and task board created but not all team members have joined and contributed.	GitHub repository and task board created but only one person joined, or one of these aspects not created.	No GitHub repository and task board created, or simply just created with nothing additional done.

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Sprint 2

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
User stories the suitability and completeness of the user stories so far identified.	User stories are clearly defined and are appropriate for the application. Set of user stories suitably complete for this stage of the project.	User stories are generally clear, and mostly appropriate for the application. Set of user stories may not be as complete as expected at this stage.	User stories lack some clarity, and/or may not all be appropriate for the application defined. Provided user stories are sparse.	No user stories provided, or user stories are inappropriate or mostly incomplete.
User story tracking suitable documentation and tracking of user stories using a suitable task board.	All user stories are on task board, with suitable tracking of story progress including assignment of tasks to individuals.	Most user stories are on task board, and generally tracking of progress is being undertaken but some gaps in areas, for example assignment of people.	User stories are not clearly tracked in all instances, and detail is missing in areas, such as the assignment of people to stories.	User stories are not tracked, or the level of tracking is very limited.
Software design the suitability and completeness of the software design presented given the application design proposed.	Use case diagrams that clearly map to user stories provided and provide high-quality information of the system design. Activity diagrams are of high-quality illustrating the flow of control in the application. Suitable class, state, and sequence diagrams are also provided based on the idea specified.	Use case diagrams generally map to user stories and provide good-quality information of the system design. Activity diagrams are good-quality but do not cover the complete flow of control in the application. Suitable class, state, and sequence diagrams are also provided based on the idea specified.	Use case diagrams do not map well to user stories and are lacking in some detail. Activity diagrams also lack some detail on the overall flow of control in the system. Class, state and sequence diagrams are sparse.	No diagrams provided, or use case diagrams do not map to user stories. Activity diagrams not provided, do not cover much or the system flow of control, or are inappropriate. Class, state and sequence diagrams not provided or are inappropriate.

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
UI design the suitability and completeness of the UI design aspects such as wireframes and UX flow.	Wireframing and flow between identified screens clearly illustrated and clearly map to the complete set of user stories and other design aspects defined.	Wireframing and flow between screens is generally completed but lacking some detail or perhaps missing some of the actual screens. Some of the user stories are not clearly mapped in the UI design.	Wireframing and flow between screens is incomplete as defined from user stories and other design elements.	No UI elements provided, or UI design presented is unsuitable for the application defined.
Team collaboration the evidence of the team functioning and working collaboratively to complete the work defined.	Clear evidence that the team is working collaboratively on the project as evidenced in the deliverables, the task board and GitHub repository.	Evidence that the team is generally working collaboratively although the task board and GitHub repository indicate that there are some inconsistencies between team members work.	Evidence provided doesn't indicate that the team is collaborating very effectively to deliver the project.	No evidence indicating that the team is working together, or the level of collaboration is insufficient to support successful project delivery.

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Sprint 3

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
User story tracking suitable documentation and tracking of user stories using a suitable task board.	All user stories are on task board, with suitable tracking of story progress including assignment of tasks to individuals.	Most user stories are on task board, and generally tracking of progress is being undertaken but some gaps in areas, for example assignment of people.	User stories are not clearly tracked in all instances, and detail is missing in areas, such as the assignment of people to stories.	User stories are not tracked, or the level of tracking is very limited.
Technical progress the degree of progress the team has made in the delivery of their technical requirements.	Excellent progress made towards delivering a final version of the product based on the current project and capabilities of the team.	Team has made satisfactory progress towards delivering their final product version based on the current project and capabilities of the team.	Team has not made satisfactory progress towards delivering their final product version based on the current project and capabilities of the team.	No progress made, or progress is minimal at best.
Database integration the completeness of the database implementation and integration into the application.	Database has been fully designed and implemented to support the project. All necessary data for the project has been entered.	Database has been fully designed and implemented to support the project. Only minimal data added at this point.	Database is not yet fully designed and implemented. Therefore, only minimal data is available.	Database has not been designed and implemented, or has very minimal design and implementation.
Template development the progress of the templates and understanding of how they integrate with the backend	Templates for all features developed and dynamic data pulling through	Some templates developed with some dynamic data correctly pulling through	Not all templates fully defined and implemented. Templates don't show correct data from database.	No templates defined, or very minimal set produced with no connection to the database.
GitHub repository practice evidence of team	Obvious evidence that the team is collaborating well,	Evidence that the team is collaborating, and in general	Some evidence of the team collaborating. Not as clear	No or little evidence of the team collaborating. Team is

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
collaboration and good practice in GitHub.	everyone is committing to the project, and commits are frequent and small.	everyone is committing to the project. Commits are not always frequent and small.	that the whole team is committing to the project. Commit practice needs work.	not using GitHub universally. Commit practice is non-existent or poor.
Team collaboration the evidence of the team functioning and working collaboratively to complete the work defined.	Clear evidence that the team is working collaboratively on the project as evidenced in the deliverables, the task board and GitHub repository.	Evidence that the team is generally working collaboratively although the task board and GitHub repository indicate that there are some inconsistencies between team members work.	Evidence provided doesn't indicate that the team is collaborating very effectively to deliver the project.	No evidence indicating that the team is working together, or the level of collaboration is insufficient to support successful project delivery.

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Sprint 4

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
User story tracking suitable documentation and tracking of user stories using a suitable task board.	All user stories are on task board, with suitable tracking of story progress including assignment of tasks to individuals.	Most user stories are on task board, and generally tracking of progress is being undertaken but some gaps in areas, for example assignment of people.	User stories are not clearly tracked in all instances, and detail is missing in areas, such as the assignment of people to stories.	User stories are not tracked, or the level of tracking is very limited.
Technical progress the degree of progress the team has made in the delivery of their technical requirements.	Excellent progress made towards delivering a final version of the product based on the current project and capabilities of the team.	Team has made satisfactory progress towards delivering their final product version based on the current project and capabilities of the team.	Team has not made satisfactory progress towards delivering their final product version based on the current project and capabilities of the team.	No progress made, or progress is minimal at best.
GitHub repository practice evidence of team collaboration and good practice in GitHub.	Obvious evidence that the team is collaborating well, everyone is committing to the project, and commits are frequent and small.	Evidence that the team is collaborating, and in general everyone is committing to the project. Commits are not always frequent and small.	Some evidence of the team collaborating. Not as clear that the whole team is committing to the project. Commit practice needs work.	No or little evidence of the team collaborating. Team is not using GitHub universally. Commit practice is non-existent or poor.
Team collaboration the evidence of the team functioning and working collaboratively to complete the work defined.	Clear evidence that the team is working collaboratively on the project as evidenced in the deliverables, the task board and GitHub repository.	Evidence that the team is generally working collaboratively although the task board and GitHub repository indicate that there are some inconsistencies between team members work.	Evidence provided doesn't indicate that the team is collaborating very effectively to deliver the project.	No evidence indicating that the team is working together, or the level of collaboration is insufficient to support successful project delivery.

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
Application quality the overall quality of the application, based on its robustness and user interfaces.	Product is of an excellent quality, with a well built user interface and no obvious errors occurring when running the application.	Product is of a good overall quality, with generally good user interfaces and few obvious errors when running the application.	Product user interfaces have some issues, and the application has some obvious errors that lead to instability.	User interface is poor and does not meet the requirements set out for the project. Application often fails or lacks stability.
Code quality ability to format and document code for human consumption (Good Style)	Program contains appropriate documentation for all major functions, variables, or non-trivial algorithms. Formatting, indentation, and other white space aids readability.	Program contains some documentation on major functions, variables, or nontrivial algorithms. Indentation and other formatting is appropriate.	Program contains some documentation (at least the student's name and program's purpose) but has occasionally misleading indentation.	Program contains no documentation, or grossly misleading indentation.

Module name: Software Engineering (CMP020N204S)

Lecturer: Dr Sobhan Y. Tehrani

Coursework

Individual Contribution

Criteria	Excellent	Satisfactory	Not Satisfactory	Not attempted
Presentation how well organised is the report.	Good organisation, points are logically ordered, report order flows together.	Organised, although some points are somewhat jumpy. General flow of report works.	Some organisation points jump around. Flow is unclear.	Poorly organised. No logical progression. Overall flow is vague.
Discussion overall level of discussion in the report.	Excellent overall level of discussion provided.	Good discussion level provided although unclear or inconsistent in a couple of places.	Discussion is good but lacks clarity and consistency in places.	Discussion is poor and does not provide clarity to the report.
Quality of information ability to provide details to support the report.	Supporting details specific to subject.	Some details are non-supporting to the subject.	Details are somewhat unclear and do not support the topic.	Specific details undefined or difficult to find.
Depth of reflection ability to provide an in-depth analysis of the overall process of the project.	Excellent depth of analysis on the overall process undertaken in the project including the product delivered based on the original vision.	Good depth of analysis, but could have gone into more detail of the product delivered based on the original vision.	Analysis is brief and doesn't really explore how the developed product compares to the original vision.	No reflection or little reflection on the overall project process and the product delivered.
Critical self appraisal ability to critically analyse own contribution to the project process based on role.	Excellent level of self-appraisal, highlighting what went well and what would have been done differently in hindsight based on the project delivered.	Good level of self-appraisal, including some analysis of what went well and what could have gone better given hindsight.	Self-appraisal is brief and doesn't go into any real analysis of what went well or what could have gone better.	No self-appraisal or little self-appraisal on own performance in the team.
Critical team appraisal ability to critically analyse other team members' contribution to the project process based on role.	Excellent level of appraisal of the team's work and how they have contributed to the delivery of the project, including individual roles undertaken in the project.	Good level of appraisal of the team's work and how they have contributed to the delivery of the project, although might be lacking exploration of individual roles.	Team appraisal is brief and doesn't really analyse the individual roles within the team.	No team appraisal or little team appraisal and how they have contributed to project delivery.