INFO30005 2021
Deliverable 4 - Requirements

This document describes the final deliverable arising from your semester-long group project.

Each group should deliver:

1. Your completed web application (software, 40%)
2. A report on your work (PDF document, 5%)

Each individual student should deliver:

3. A statement on your group member contributions (Canvas form, 0%)

This deliverable is due on the last day of week 12, Friday 28th May 2021, by midnight.

1. Web application (40%)

Your app should offer the features described in the Requirements document that was handed out at the start of semester. Aim to get all of the customer and vendor features working.

Your customer UI needs to be responsive so that it is usable on both phone and desktop. Your vendor UI needs to be usable on tablet. We will test your app using these screen sizes:

- Phone: 375 x 812 (iPhone X, portrait)
- Tablet: 1024 x 768 (iPad, landscape)
- Desktop: 1920 × 1080 (monitor, landscape)

The following items will be assessed:

- Your live server (your app deployed on Heroku)
- The repo(s) in which your app code is stored. (staff will access this in GitHub)
- A zip file of your code. If you have two separate repos, submit one zip file per repo. (submit your zip files to Canvas)

Your repo must contain a README file which contains:

- The commit id that you want us to mark.
- The URL of your live website on Heroku.
- The connection string (including login and password) to your database on Atlas.

Write unit and integration tests for the following feature:

- "Using the vendor app, the van operator sets the status of their van."
- The integration tests should check the routes, controllers and models implemented to support the above feature.
- The unit tests only need to check the controller functions associated with the above feature.

Provide the details of how to run the tests in your README file.

Customer Password Policy: We want groups to apply the following policy to the customer passwords: that is, make sure that the password that a customer supplies when they register has:

1. At least one alphabet character (upper or lower case A-Z)
2. At least one numerical digit (0-9)
3. A length of at least 8 characters

Ensure that this password policy is checked before passwords are stored in the database.

Passwords MUST be hashed before storing them in the database.

These web-application items are together worth 40% of your total semester marks.

2. Report (5%)

Write between 1000 and 2000 words on the following topics:

1. A description of your development process:
   - Any tools, libraries, frameworks and code sources you used, what you took from them, and your reasons for choosing them.
   - Each team member's role.
   - Your team's process for assigning tasks and managing the repository.
2. A description of your system architecture:
   - Use diagrams and text to illustrate how the different components are interconnected.
   - Include a diagram to illustrate your database design. Use a UML class diagram (if you are familiar with UML) or an informal diagram like figure 1. (Figure 1 is based on the MongoDB lecture).
   - You may show brief code snippets to illustrate interesting aspects of your implementation.

This report is worth 5% of total semester marks.

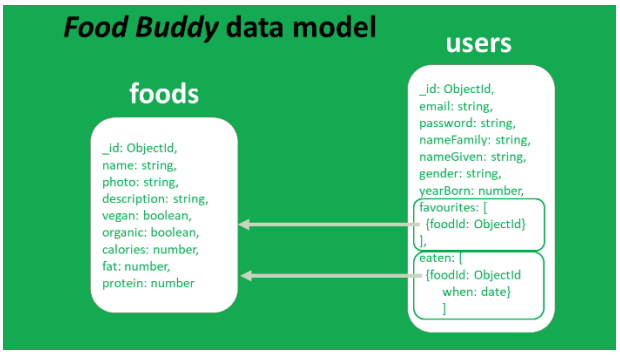

Figure 1: informal data-model diagram

3. Group member contributions

Each individual student should submit an assessment of the contributions of their group members to the project. Do this via the form on Canvas. It looks like this:

Group member name	Contribution %	Comment

Describe the contribution of each member of your group, including yourself. Estimate the contribution size, and write a comment summarising what that group member did.

Ideally, groups will have equal contribution from all members. For example, in a five-person group, each member should contribute 20% of the project work.

We will provide a submission link titled “Group Contributions” on Canvas for students to enter this information. Deliverable 4 and the assessment of contributions are due on the same date.

Marking Rubric

Software

Criteria	Excellent	Good	Inadequate
Features 20%	All relevant functionalities are fully built and working properly.	The core usage scenarios are mostly implemented, but there are still large gaps.	The content of the core usage scenarios partially load from an appropriate database.
User Interface 10%	Professional look and feel, excellent use of colour, type, and layout. Very easy to use, with no glitches.	The interface looks fine, but could still be substantially improved. Confusion with interface.	Inadequate use of colour, type, and layout. Severe usability problems.
Code quality 10%	A high standard of coding and commenting is evident, the code is easy to read and maintain. Git history.	Reasonable clarity, code can be followed with some effort, comments are helpful, logical organization lacking .	Extremely hard to read, little or no attention to code formatting, lack of logical organization, little commenting.

The 4 'bonus' features mentioned in the Requirements can receive up to 2% each.
However bonus marks cannot raise your total mark for this section above 40.

Report

Criteria	Excellent	Good	Inadequate
Description of Process 2%	Report provides sufficient details on technology choices, member contributions, and project management strategies.	Report provides some details on technology choices, member contributions, and project management strategies.	Report provides very brief or no details on technology choices, member contributions, and project management strategies.
Description of Architecture 2%	Detailed discussion of functionality supported by appropriate visuals. Architecture is explained clearly.	Significantly lacking discussion of functionality or architecture. Limited use of visuals.	Very brief discussion of functionality or architecture.
Report organisation and clarity of expression 1%	Up to the standard expected of a good conference paper or business report. Well organised. Pleasant to read.	Some defects, including completeness, structural oddities or difficult passages. Difficult to read.	Difficult to read throughout. Important material is frequently missing. Little to be gained from reading the report.