Wednesday, 25 August 2021

Assignment 2

COMPSCI 235 – SOFTWARE DEVELOPMENT METHODOLOGIES
(Semester 2, 2021)

SCIENCE

Brief

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND
SCIENCE

- The objective of this **two-student group** assignment is to design and build a Web application that lets users interact with a collection of books
  - This assignment builds on Assignment 1 in that you can incorporate aspects of your already developed domain model into the Web application
- You have **considerable freedom** in this assignment
  - There is no prescriptive specification; rather, there is a general statement of requirements that you should interpret
  - Make your own decisions as to how you'd like to implement the requirements
- You **choose what grade** you want to aim for
  - There are A+ to C grades, with increasing requirements to satisfy the higher grades
- You need a Github account (as created in the second lab), and you will receive a **code skeleton** through a **Github Classroom** repository, which is also where the groups will be formed and where you finally submit this assignment (see Github Classroom instructions here)

Group work

You will work in **groups of two** students on this assignment. This will help you distribute workload and will enable you to practice working on a common code repository in a small team.

Note: If students prefer to work alone on this assignment, this will be an option given the lockdown situation.

Group Marking Criteria -
Each member of the group/team will submit a confidential allocation of percentage of work done by them and their groupmate. This will be submitted through a Google Form, with the link to be published closer to assignment deadline. Your mark for this assignment will reflect your contribution in the assignment working.

FAQs

- Questions relating to the assignment will be answered by the teaching team.
  - You should post your questions on Piazza
- A Canvas page is dedicated to frequently asked questions -
  - This page will be updated over the duration of the assignment, capturing the results of discussions with the teaching team
  - Refer to the FAQ on the Canvas page before posting on Piazza, in case your question has already been answered
  - The Canvas FAQ page is available here

	C grade	B grade	A grade	A+ grade
% range	50 - 65	66 - 80	81 - 90	91 - 100
Functional requirements	• Browsing books	• Displaying/searching books based on authors, release years, publishers • Registering, logging in/logging out users • Reviewing books	• A new cool feature	
Non-functional requirements	• Conformance to established project structure • Effective use of HTML, CSS and Jinja • Appropriate use of HTTP • Application of Repository pattern • Unit and integration testing of your developed code!	• Use of Blueprints • Use of authentication techniques • Use of HTML forms / WTForms		• Cool feature design report

Grading factors (C)

- **Functional requirements**
  - Minimally, the Web application should allow information about books to be displayed in a Web browser
  - Users should be able to navigate books, noting that there is a limit on the number of books that can be displayed on a single Web page
- **Non-functional requirements**
  - Conformance to the project structure used for sample Flask applications (as also discussed in labs)
    - Inclusion of requirements.txt file, readme file etc.
  - User interface
    - Use of CSS to style HTML pages
    - Appropriate use of Jinja templating to define HTML layout, pages and partials
  - Web interface
    - Appropriate definition of entry points (URLs)
    - Appropriate use of the HTTP protocol (e.g. query parameters, response codes etc.)
  - Testing
    - Thoroughness of unit and integration tests (this requirement also applies for grades B to A+ accordingly)
    - Presence of unit tests for all key components (domain model, repository, any service layer components)
  - Application of the Repository pattern (Memory repository)
    - Design of a suitable interface and implementation for querying data about books

Grading factors (B)

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND
SCIENCE

- For B grade submissions, all functional and non-functional requirements for a C grade submission must be met, **and in addition**, these requirements also have to be met:
  - **Functional requirements**
    - Users should be able to search for books by selecting authors or publishers, or by specifying e.g. a release year
    - Users should be able to register with the application, and login and logout
    - Logged in users should be able to provide reviews or ratings for books
  - **Non-functional requirements**
    - Use of Blueprints
      - Blueprints should be used to appropriately separate areas of application functionality and adhere to the principle of Single Responsibility
    - Authentication
      - Signed cookies should be used in addition to ensuring that only logged-in users can write reviews
    - Use of HTML forms
      - Forms should be used to allow users to select filtering options for displaying books by author or publisher or other criteria
        - » Use of WTForms is recommended

Grading factors (A)

- The University of Auckland's assessment policy [1] defines what's expected of A grade work, and includes
  - *Work of high to exceptionally high quality showing excellent knowledge and understanding of subject matter and appreciation of issues*
  - *High level of creative ability, originality and critical thinking*

[1] UoA assessment policy

Grading factors (A)

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND
SCIENCE

- For A grade submissions, all functional and non-functional requirements for B and C grade must be met, **in addition**, these requirements also have to be met:
  - **Functional requirements**
    - A cool, non-trivial, new feature of your choosing should be implemented
      - Possible features include, but are certainly not limited to:
        - » A recommendation system for books based on user reviews and/or preferences
        - » Extending the application to work with an inventory of books, including prices and number of items in stock (i.e. tailoring the application towards a web-shop)
        - » Distinguish between available books in the library, and users' favourite books that could be put into a reading list
        - » You may want to have a look at https://www.goodreads.com for further inspiration
        - » … (we are highly interested in your own individual ideas!)
  - **Non-functional requirements**
    - Nothing extra, but feel free to use any Python libraries that were not yet introduced in COMPSCI 235
      - But please note: **You shouldn't use SQLAlchemy** or any sort of database library, since database-based repositories will be covered in Assignment 3!

Grading factors (A+)

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND
SCIENCE

- For A+ grade submissions, all requirements for B, C and A grade submissions must be met, **and in addition**, these requirements also have to be met:
  - **Design report**
    - A report that introduces your cool new feature, and which describes key design decisions with justification that you have made in developing your project. You should refer to any design principles and patterns applied and the benefits they provide in the context of your project
    - A report of 2-3 pages in length should be sufficient
    - There is a small gap between an A grade and an A+ grade, so if you meet the requirements for A, write the report to get an A+!

Submission and due date

- You should submit:
  - Your code to your specific Github Classroom team repository for Assignment 2
  - A+ submissions: an additional design report as a **pdf** located in the root folder of your repository
- Submitted projects must conform to the project structure for Flask applications presented in COMPSCI-235
  - Projects must include a readme file, a requirements.txt file and a wsgi.py file
- When assessing your work, markers will expect to:
  - Find a readme file in the project's root directory that explains how to set up a virtual environment and install any dependencies via a requirements.txt file
  - Run the application by typing 'flask run', from within the virtual environment in a terminal window
  - Run all tests by typing 'python –m pytest –v tests', from within the virtual environment in a terminal window
  - **Failure to meet these requirements will result in a fail grade**
- Submissions are due **Wed, 22 September 2021 at 23:59 hrs**
- This assignment is **worth 20% of your final grade**.

Late submission penalty

- Submission by **Wednesday, 22 Sept 23:59**: **no penalty**
- Submission by **Thursday 23 Sept 23:59**: **20% penalty**, your submission will be marked out of 20 points. Your result will be multiplied by 16/20 for the final marks on Canvas.
- Submission by **Friday 24 Sept 23:59**: **40% penalty**, your submission will be marked out of 20 points. Your result will be multiplied by 12/20 for the final marks on Canvas.
- Submission by **Saturday 25 Sept 23:59**: **60% penalty**, your submission will be marked out of 20 points. Your result will be multiplied by 8/20 for the final marks on Canvas. For example, if your submission is assessed giving 12.5 out of 20 points, and is submitted on Saturday, you will receive 5 marks for it.
- Submission after **Saturday**: **not possible, 0 marks** for the assignment

Note: You have to inform us of your delay before 22 September. It is not possible to submit one version on 22 Sep and an updated version later, you have to decide on one submission!

Advice

- **Start** this assignment **upon** its **release** and work on it consistently
  - You can't expect to pass this assignment by starting it late
- Put into practice Agile practices
  - Start with a C-grade submission and iteratively work towards the grade that you want, submit your progress frequently to GitHub
  - As practiced in Assignment 1, use test-driven development – it's effective, really!
    - Write your tests first, then write your application code
  - Decompose large development items into small development tasks and work on these tasks using e.g. a Kanban board to track progress
    - This way, you'll make progress and avoid having to work on the whole problem at once – rather than becoming overwhelmed
- Challenge yourself, do the best you can
  - Produce something you'd be proud to show your grandma, or to a future employer!
  - Don't ask questions like "what's the minimum I need to do for a pass?"
- Use the provided COVID-19 template application for reference
  - It illustrates all that's required for A to C grades
- Use external resources to supplement lecture materials
  - Use official documentation
    - Flask https://flask.palletsprojects.com/en/1.1.x/
    - Jinja https://jinja.palletsprojects.com/en/2.11.x/
    - WTF forms https://flask-wtf.readthedocs.io/en/stable/, https://flask-wtf.readthedocs.io/en/stable/
    - Python https://www.python.org/ (please use a recent version of Python (>= 3.6), we have tested on Python 3.7 and 3.8!)
  - Pay attention to up-voted questions and responses on Stack Overflow – they indicate credibility
  - Use other resources of good repute, e.g. https://realpython.com/
  - Share good finds with fellow students on Piazza
- Don't hack, take a disciplined approach to developing software
  - Use pen and paper to sketch out your design and to think it through before coding, use a journal to
  - Draw diagrams of data structures, web page layouts and the software architecture – visualising what you want to implement really helps you to understand what you're going to code

References

- GitHub Classroom instructions -
  - https://canvas.auckland.ac.nz/courses/60516/pages/github-classroom-instructions
- COVID’19 template web application code -
  - https://github.com/martinurschler/2021CompSci235-03-CovidWebApp
- Frequently asked questions -
  - https://canvas.auckland.ac.nz/courses/60516/pages/assignment-two-frequently-asked-question-2021