

Assignment 3

COMPSCI 235 – SOFTWARE DEVELOPMENT METHODOLOGIES



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

SCIENCE

- The objective of this assignment is to modify/extend the web application concerned with a collection of books in a library, such that instead of a memory repository, we now use a SQLite database to persist data.
 - This assignment builds on Assignment 2, so you need a working version of your Library application in order to extend it further.
 - Please get into contact with us by mail if you don't have a working version of A2.
- All the features that you implemented in Assignment 2 should now be switched to work with a database instead of an in-memory repository.

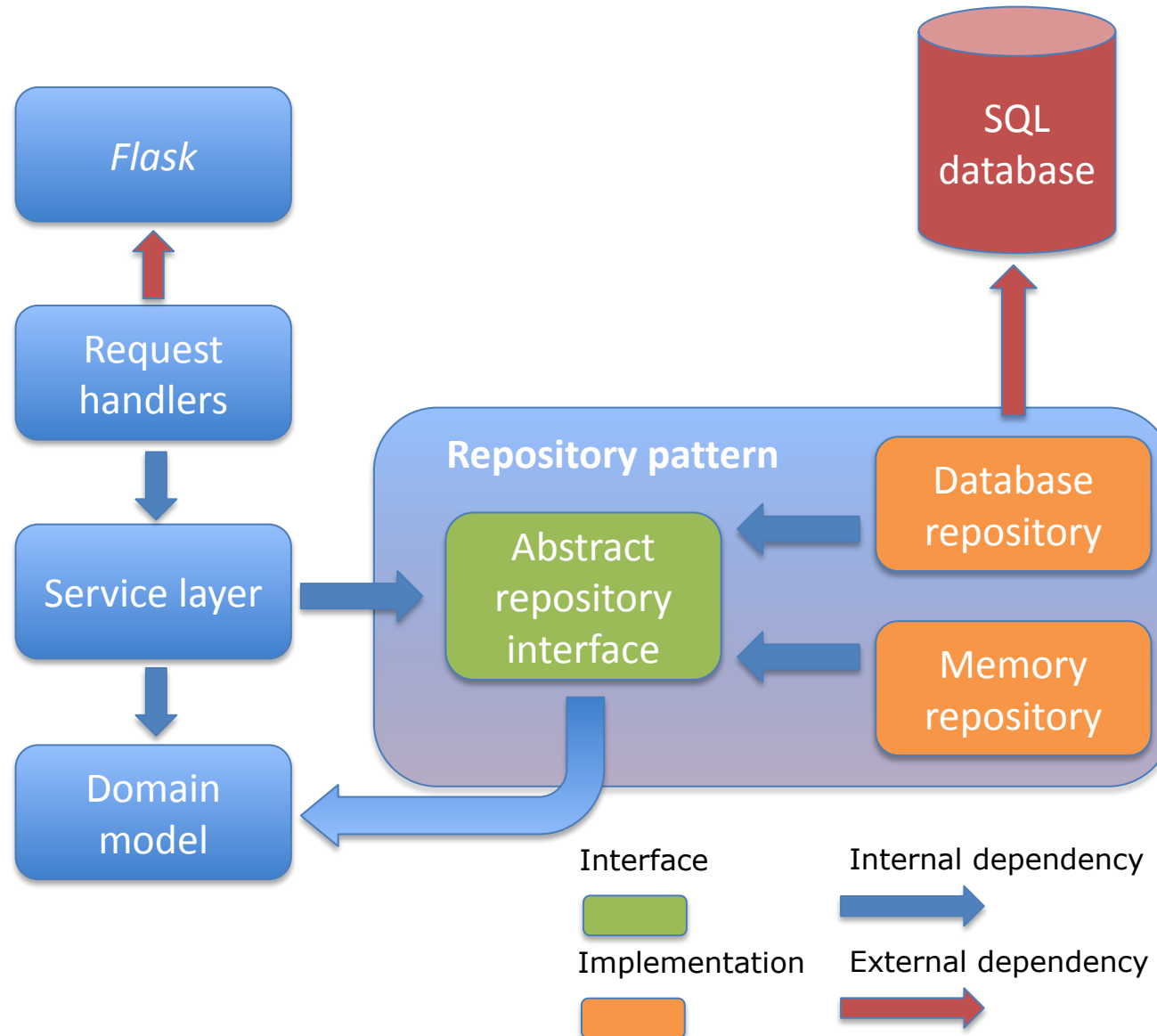
Our application



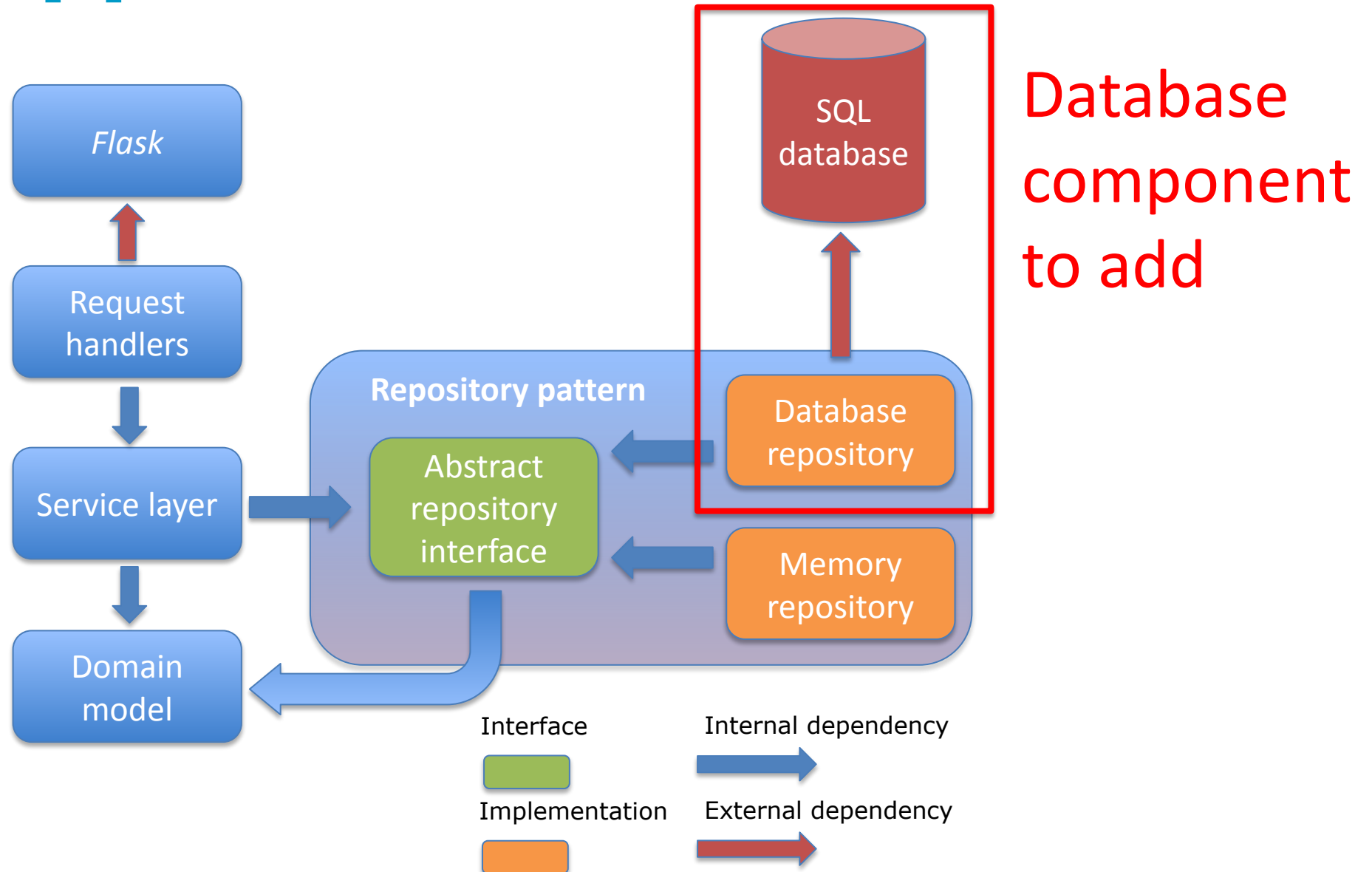
SCIENCE

- At this stage you should have a running web application that allows you to e.g. browse books, login users, write book reviews, ..., served by the memory repository.
- Now we will replace the data stored in memory with a “simple” file based SQLite database.
- This can still be seen as an initial development step, compared to a dedicated external database server e.g. running MySQL, or Oracle.

Our application



Our application



Assignment 3

How to go from memory-based storage to persistent database?

We will use Python **SQLAlchemy**. The required steps are:

1. Include a **database engine** and do the setup and configuration to initialize the database connection and to allow communication.
2. Create the **object relational mapper** that connects our domain model with the relational database.
3. Write code that **populates** our database with data from external (e.g. json) files.
4. Write the **SqlAlchemyRepository** class, derived from AbstractRepository, to enable the communication with the application through the service layer.
5. Write **test cases** for the database aspects of the application.

Assignment 3

Make an update of our provided COVID-19 web application from here: <https://github.com/martinurschler/2021CompSci235-03-CovidWebApp> using 'git pull' or clone it if you haven't yet done so.

- We have added a new branch 'SQLite_Database_Repo'
- You can have a look at the local branches using 'git branch' (note that 'git branch -a' shows local and remote branches) and you can switch to this branch using 'git checkout SQLite_Database_Repo'
- This new branch implements the SQLite based database repository for the COVID application.

Grading

- This assignment will **lead to a total of 12% of marks.**
- Grading requirements:
 - The web application works with at least the **same** features as in Assignment 2, but now using the persistent storage in a database. Therefore, the information displayed on the web page has to be fetched from the database. If applicable, users and reviews also have to be persisted in the database. We will inspect the app and the database file when marking!
 - There are tests available that demonstrate the functioning of the database similar to the memory based web application.

	C grade	B grade	A grade	A+ grade
% range	50 - 65	66 - 80	81 - 90	91 - 100
Functional requirements	<ul style="list-style-type: none"> Browsing books 	<ul style="list-style-type: none"> Displaying/searching books based on authors, release years, publishers Registering, logging in/logging out users Reviewing books 	<ul style="list-style-type: none"> A new cool feature 	
Non-functional requirements	<ul style="list-style-type: none"> 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! 	<ul style="list-style-type: none"> Use of Blueprints Use of authentication techniques Use of HTML forms / WTForms 		<ul style="list-style-type: none"> Cool feature design report

Grading

- If you submitted a C/B range application for Assignment 2, and if you transfer the same features using a database, you will qualify for a C/B range grade in Assignment 3.
- If you already submitted a A/A+ range application for Assignment 2, and if you transfer the same features using a database, you will qualify for an A/A+ range grade in Assignment 3.
- Additionally, there's an opportunity to extend **solely** the database features to reach a higher grade for Assignment 3.

For example, if you had previously submitted a B grade application for Assignment 2, you could add a cool new feature now just for the database version to secure an A grade for Assignment 3. And in this case, you may also include a report for A+.

Group work

We would highly encourage you (and this is in your own benefit) to work in your original group as formed for A2, since A3 is a continuation of what was produced for A2. In exceptional circumstances, where there were issues in the group dynamics, you may choose to work alone or recreate your team. In this case you have to get into contact with the teaching team.

Group Marking Criteria -

Similar to A2, each member of the group/team will submit a confidential allocation of percentage of work done by them and their groupmate through a Google Form which will be published later.

Submission and due date

- Submission:
 - Please **continue using your existing team repositories from Assignment 2** for submitting Assignment 3. Please get into contact with the teaching team if you have an exceptional reason to change teams.
- 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 test_folder', from within the virtual environment in a terminal window
- Submissions are due **Friday, 15 October 2021 at 23:59 hrs**
(Late submission penalties similar to A2 will apply).

- Similar advice as for Assignment 2 should be followed.
- Use test-driven development.
- Study the material of labs from week 8 and especially week 9. Make heavy use of the COVID-19 application and. You can use code from there as freely as you want.
- We will also provide a recording of a help tutorial similar to Assignment 2.
- Use external resources to supplement lecture materials
 - SQLAlchemy <https://www.sqlalchemy.org/>
- 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, put these things into your personal journal (which will not be marked).