COMP2001
Information Management & Retrieval

20 CREDIT MODULE

ASSESSMENT: 100% Coursework **W1: 30% Set Exercises**
W2: 70% Report

MODULE LEADER: Haoyi Wang
MODULE STAFF: Martin Read

MODULE AIMS

- To introduce students to fundamental principles around graphical representation along with information management, database systems and modelling.
- To consider issues around image compression techniques, how humans can access information and data to support their needs, learn declarative queries and consider common designs for database systems.
- To understand the differences between relational and semi-structured data models and use appropriate data modelling techniques.

ASSESSED LEARNING OUTCOMES (ALO):

1. Demonstrate explicit uses of modelling techniques to gain access to information and data to support a given need.
2. Illustrate an appropriate technical solution to the problems of information privacy, integrity, security and preservation.
3. Illustrate the design of an application of moderate complexity to elicit and visualise information from a data store.

Overview

This document contains all the necessary information pertaining to the assessment of *COMP2001 Information Management & Retrieval*. The module is assessed via **100% coursework**, across two elements: *30% Set Exercises* and *70% Report*.

The two elements provide the opportunity for you to show you have met the Assessed Learning Outcomes listed above. The Set Exercises (CW1) allow you to show that you can use modelling techniques (LO1) and the Report (CW2) provides the evidence for how you have created an API providing access to data (LO2 and LO3).

The sections that follow provide the details you need but the scenario is not exhaustive – there are some aspects that you will need to complete for yourself. The seminar sessions are the best place to gain clarification, to ask questions and to make sense of incomplete data.

All assessments will be introduced in class to provide further clarity over what is expected and how you can access support and formative feedback prior to submission. Whilst the assessment information is provided at the start of the module, it is not necessarily expected you will start this immediately – as you will often not have sufficient understanding of the topic. The module leader will provide guidance in this respect.

The submission and expected feedback dates are presented in Table 1. All assessments are to be submitted as per the deliverables outlined with the principle electronic submission via the respective DLE module pages before the stated deadlines.

	Submission Deadline	Feedback
Set Exercises (30%)	4th November 15.00 GMT	3rd December
Report (70%)	7th January 15.00 GMT	5th February

Table 1: Assessment Deadlines

Task

You must imagine yourself as part of a development team that is creating a data-driven application that is made up of a number of interlinked micro-services. The exact topic of the micro-service is given below.

There are several parts to the micro-service that you will need to create

- A set of tables to store the data in a relational database
- Stored procedures to manage the data (CRUD)
- A View to combine the data from different tables
- Sample data to illustrate the view working
- API endpoints for creating, reading, updating and deleting a given resource

Scenario
The team is creating a well-being trail application. The location of the trails you use/create is your choice. The product vision is:

> *For people who wish to enjoy the outdoors, to enhance their wellbeing and to have a reason to explore a particular area, the Trail App is a full trail management application providing a reason to explore a given area.*

The application will meet the following user stories that have been placed on the product backlog

Product Backlog

- As an Administrator I wish to create a new trail with information
- As an administrator I wish to check that a new trail with information exists
- As an administrator I wish to edit the information for an existing trail
- As an administrator I wish to delete an existing trail
- As a user of the trail app I wish to select and view a particular trail using my android mobile device

An overview of the architecture is provided in the diagram below.

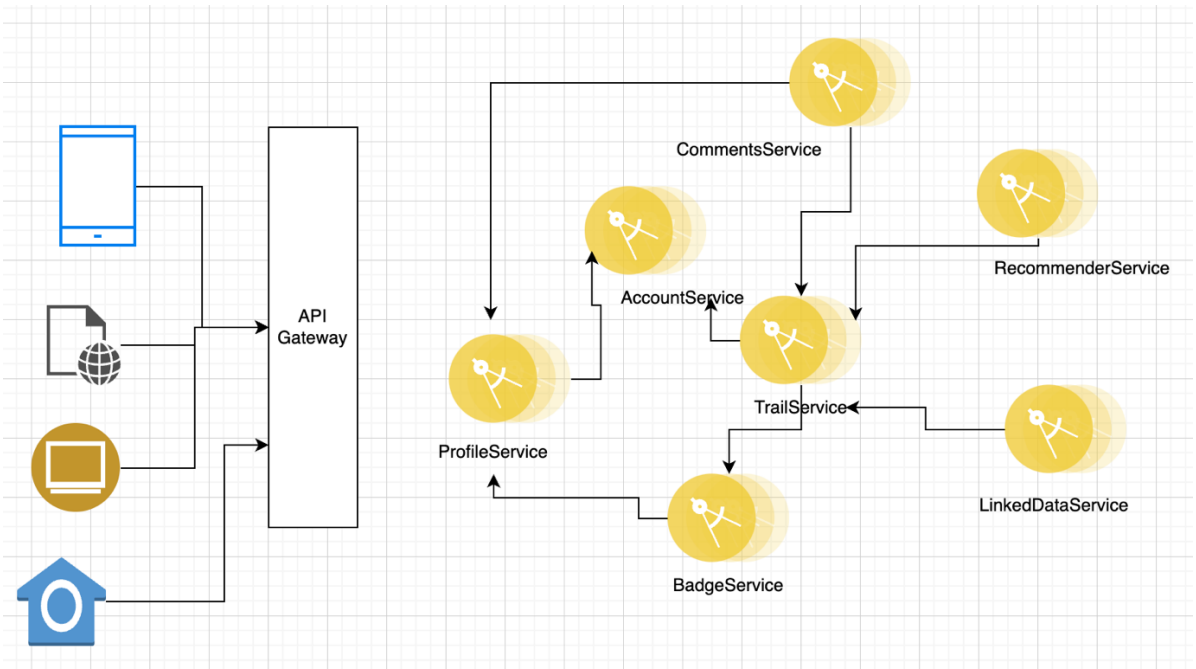

Figure 1: Architecture diagram for Trail Application

You must implement the following micro-service:

- TrailService
    - CRUD operation on trails
    - Anyone can view a trail but has limited view
    - Trails are a series of location points
    - Trails are owned by a user
    - Error checking for how far apart locations is out of scope

The application must use the existing Authenticator API found at https://web.socem.plymouth.ac.uk/COMP2001/auth/api/users

The following accounts must be used in the application. These accounts exist in the authenticator API application. The roles that these users hold is to be held in the Trail App.

User Name	Email	Password
Grace Hopper	grace@plymouth.ac.uk	ISAD123!
Tim Berners-Lee	tim@plymouth.ac.uk	COMP2001!
Ada Lovelace	ada@plymouth.ac.uk	insecurePassword

Non-Functional Requirements

- The data must be stored in a Microsoft SQL Server database hosted on dist-6-505.uopnet.plymouth.ac.uk.
- The server-side code must be hosted on the web server web.socem.plymouth.ac.uk.
- The server-side code must be written in Python.
- Response times are not within the scope of the application.
- All appropriate attempts to mitigate the OWASP top 10 vulnerabilities are to be taken where possible.
- The micro-service must conform to RESTful API principles.

Your micro-service must provide the following:

- A clear endpoint interface represented via swagger.
- Output of data in JSON format.

You are NOT to implement any other micro-service.

Sample Data

Please see https://www.alltrails.com/trail/england/devon/plymbridge-circular (you may need to copy & paste the link) as an example of sample data.

The rest of the document provides the assessment information.

Assessment 1: Set Exercises

CW1 Overview

The following set exercises assess different parts of your micro-service.

Carry out an analysis of the micro-service, identify the appropriate data items, design and then create the SQL that will store the data appropriately and preserve the integrity of the data. You are to implement your design by deploying your SQL to your allocated Microsoft SQL Server database running on dist-6-505.uopnet.plymouth.ac.uk.

Submit a PDF document containing the evidence for all the exercises given below.

Create an initial Entity Relationship Diagram (ERD) for your micro-service. Provide this, clearly labelled, in an Appendix to your report. **Failure to provide this will result in a lower mark.**

The initial ERD & any subsequent version/s (produced after resolving many-to-many relationships, adding optionality, removing redundant relationships, etc) provided in the Appendix will not be marked but may explain how your final ERD was developed.

Set Exercise 1: Normalisation: 20 marks

Carry out Normalisation to Third Normal Form (3NF) on the trail details page of a trail such as the Plymouth Waterfront or Plymbridge Circular, found on the website. Do not include conditions, reviews, photos, activities, completed, directions, printable map, share, more. **State any assumptions you have made.**

Normalise the attributes to 3NF showing all intermediate stages, namely: Un-normalised Form (UNF), First Normal Form (1NF) and Second Normal Form (2NF). Be careful to identify the attributes used for keys.

Additional marks will NOT be gained for optimising the results of 3NF, though you may still find it worthwhile to optimise your results. You must name the 3NF relations.

Draw the partial ERD using the taught notation of soft boxes and crows feet. This should be included at the end of the Set Exercise 1 report section, clearly labelled. This partial ERD will be assessed as part of Set Exercise 2.

Set Exercise 2: Final Entity Relationship Diagram (ERD) : 15 marks

Create a final Entity Relationship Diagram (ERD) for your micro-service.

This final ERD will be the result of combining the initial (or subsequent) ERD/s with the partial ERD resulting from your Normalisation.

This final ERD should only include entities and relationships (including relationship link phrases) that can be justified from reasonable assumptions. Make sure that all one-to-one and many-to-many relationships are resolved prior to submission. **State any assumptions you have made**.

Ensure that the ERD offers an accurate representation of the database. Inconsistencies between the ERD and the actual database implemented will mean marks are not earnt.

The next exercise moves from database design to implementation.

The result of your Database Design will be at least 3 entities & possibly many more. This would involve a lot of work to produce the SQL to create & manage multiple tables, so you might not wish to implement all of them – but you do need to implement at least 3 entities (including a 'link entity') that could be used to substantially create the trails webpage, give reasons for the tables chosen.

Implement your SQL code to create tables, views, procedures and triggers on your hosted Microsoft SQL Server database at dist-6-505.uopnet.plymouth.ac.uk.

Set Exercise 3: Database design: 15 marks

Include a completed Field Definition grid for the Attributes of each Entity to be implemented. They should be provided in an Appendix to your report.

This will help you in your implementation & justify your choices.

Set Exercise 4: SQL: 15 marks

Following your analysis, create the appropriate SQL to form at least 3 tables (including a 'link entity'), include any keys/constraints, and implement these on your hosted Microsoft SQL Server database at dist-6-505.uopnet.plymouth.ac.uk. Add sufficient data to each table to enable testing to take place.

Screenshots should be included in your main report to demonstrate (using SELECT statements) that sufficient data is present in all tables.

You must use the schema "CW1" for your implementation to differentiate your coursework tables from any others. **Ensure you have demo data within the tables. Failure to do so will result in a lower mark.**

Set Exercise 5: View: 10 marks

Identify and implement an appropriate view which could be used to substantially form the trails webpage, combining the data from at least two tables. The SQL code and screenshots to demonstrate successful implementation should be included in your main report.

You must implement the view under the schema "CW1". Remember that views can be queryable. **Ensure you have sufficient demo data within the tables so that the view returns data. Failure to do so will result in a lower mark.**

Set Exercise 6: Stored procedures: 15 marks

Identify and implement all the stored procedures needed to handle the insert, read, update and delete functionality (CRUD) for one of the tables. The SQL code and screenshots to demonstrate successful implementation should be included in your main report.

You must implement the stored procedures under the schema "CW1". Ensure you provide screenshots (before and after) providing evidence of testing your stored procedures.

Set Exercise 7: Trigger: 10 marks

Implement a trigger to automatically log the addition of a new trail. A separate log table should be used to store the details like who added which trail and the timestamp of when it was added. The SQL code and screenshots to demonstrate successful implementation should be included in your main report.

You must implement the trigger under the schema "CW1". Ensure you provide screenshot evidence of testing your trigger.

Deliverables

Submit the answers to these set exercises on ONE PDF document with clear titles for each exercise. Ensure the SQL has been deployed to your hosted Microsoft SQL Server database.

- Note your SQL Scripts must match the implementation on the module hosted Microsoft SQL Server EXACTLY.

Assessment 2: Report on micro-service implementation

CW2 Overview:
Part 2 of your coursework is to build upon CW1 to design and implement the micro-service. This will require you to draw up appropriate UML diagram/s, create the code to implement the design and deploy the code so that it can be seen to run.

SQL implemented in CW2 should be implemented under the schema "CW2". This will mean repeating your SQL implementation from CW1. This would be a good opportunity to refine your answers if required. You may find that as you implement your micro-service and decide the schema needs to be revised.

Within your design you must ensure you are showing how you are handling issues around information privacy, integrity, security and preservation. This will need to be written up explicitly in your report.

Once the tasks have been completed, you are to write up and report upon your results.

Deliverables
Your source code must be held in a GitHub repository, which you need to create. Please ensure the repo is laid out appropriately and that you are demonstrating good practice in your use of the repo. You must include a link to the repo in your report, which must be shared with the accounts *mjread* and *haoyiwang25* and ensure the account shared with has the permissions to read/view the code. You have to use your real name for the Github account (not nicknames). Do NOT try to upload your code via the DLE. Not using GitHub appropriately will limit your marks.

You MUST adapt your GitHub readme file to represent YOUR work. Failure to adjust the readme file will result in marks being lost.

Your deployed code must be hosted on your server folder, web.socem.plymouth.ac.uk. Lack of deployed code to run properly on the server folder will result in seriously limited marks.

Submit your report in PDF format. Your report must be no more than 2000 words, please use screenshots and links to code files to illustrate functionality where appropriate and UML diagrams to illustrate the design.

The report must contain the following sections:

1. Introduction (approximately 2 paragraphs). Introduce the document and signpost the reader to what they will find in it. Provide links to your GitHub repository and your hosted micro-service.
2. Background. Explain here the micro-service you implemented. Make sure you include an overview of the Trail Application and mention how the micro-service is related to the app.
3. Design. Provide the UML diagrams for the models for the micro-service. You should be including a logical ERD having evolved it from the set exercises/feedback.

4. Legal, Social, Ethical and Professional (LSEP). Discuss LSEP, including how you addressed issues around information privacy, integrity, security and preserving the data in how you designed the micro-service. You may need to refer to the earlier section of design to provide clear communication of your activities.
5. Implementation. Discuss how you implemented the micro-service. Link to code samples where appropriate.
6. Evaluation. Show here how you tested your implementation and clearly indicate areas for further work. Reflect on your strengths/weaknesses and how you might have improved your coursework implementation.

Assessment Criteria:

The report, associated diagrams and code are graded in the following way

Criteria	Description	Weighting
Introduction	Provides an introduction to the document and signposts the reader to what they will find in the document. GitHub link provided. GitHub repo in the correct place (GitHub classroom provided for module). Link provide to web.socem deployment of micro-service.	6
Background	Outline provided for the micro-service.	4
Design	Appropriate UML diagrams used. Clear demonstration of understanding how to design micro-service in place. Design clearly shows progression of understanding through high level overview through to detail.	30
LSEP	Literature used to justify appropriate for information security, privacy, integrity and preservation. Literature provided using Harvard style referencing. Appropriate approaches used to meet requirements for a secure, private application. Data items designed in a way that enforces integrity, privacy and security.	20
Implementation	Implementation discussed at appropriate level in report. Implementation matches code in GitHub classroom and provided design diagrams. Implementation meets requirements alluded to above (CRUD) Implementation is RESTful API Endpoints appropriately represented in documentation, design and implementation.	30
Evaluation	Clear evidence of testing provided. Reflection provided on further work. Honest reflection provided on weak areas of implementation.	10

General Guidance

Extenuating Circumstances

There may be a time during this module where you experience a serious situation which has a significant impact on your ability to complete the assessments. The definition of these can be found in the University Policy on Extenuating Circumstances here:
https://www.plymouth.ac.uk/student-life/your-studies/essential-information/exams/exam-rules-and-regulations/extenuating-circumstances

Plagiarism

All of your work must be of your own words. You must use references for your sources, however you acquire them. Where you wish to use quotations, these must be a very minor part of your overall work.

To copy another person's work is viewed as plagiarism and is not allowed. Any issues of plagiarism and any form of academic dishonesty are treated very seriously. All your work must be your own and other sources must be identified as being theirs, not yours. The copying of another persons' work could result in a penalty being invoked.

Further information on plagiarism policy can be found here:

Plagiarism: https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations/plagiarism

Examination Offences: https://www.plymouth.ac.uk/student-life/your-studies/essential-information/exams/exam-rules-and-regulations/examination-offences

Turnitin (http://www.turnitinuk.com/) is an Internet-based 'originality checking tool' which allows documents to be compared with content on the Internet, in journals and in an archive of previously submitted works. It can help to detect unintentional or deliberate plagiarism.

It is a formative tool that makes it easy for students to review their citations and referencing as an aid to learning good academic practice. Turnitin produces an 'originality report' to help guide you. To learn more about Turnitin go to:
https://help.turnitin.com/feedback-studio/turnitin-website/student/quickstart.htm

Referencing

The University of Plymouth Library has produced an online support referencing guide which is available here: http://plymouth.libguides.com/referencing.

Another recommended referencing resource is Cite Them Right Online; this is an online resource which provides you with specific guidance about how to reference lots of different types of materials.

The Learn Higher Network has also provided a number of documents to support students with referencing:
References and Bibliographies Booklet:
https://aldinhe.ac.uk/product/learnhigher-resources/references-and-bibliographies-booklet/
Checking your assignments' references:
https://aldinhe.ac.uk/product/learnhigher-resources/checking-your-assignments-references/

Responsible Use of Artificial Intelligence (AI) in Assessments

While technological advances such as AI (including ChatGPT) can be useful in supporting academic work, e.g. by aiding brainstorming, students are expected to use these responsibly and within the boundaries of academic integrity.

In general, AI tools such as ChatGPT should not be used in generating the final version of your work for submission. Utilising ChatGPT or similar AI tools to generate the assessed work that you submit is considered plagiarism and a breach of our university's academic offences regulations. Such practices can result in disciplinary action, up to and including a requirement to withdraw from the university. You can find the current version of academic regulations and related guidance here: https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations.