Cardiff School of Technologies

Assessment Brief

Module Code	Module Title
CIS6003	Advanced Programming
Academic Year	**Semester**
23-24	1

Module Leader email

crodriguezrivero@cardiffmet.ac.uk

Content

Assessment Details 3
Submission Details 6
Assessment Criteria 7
Further Information .....错误！未定义书签。
Who can answer questions about my assessment?错误！未定义书签。
Referencing .....错误！未定义书签。
Submission problems .....错误！未定义书签。
Unfair academic practice .....错误！未定义书签。

Assessment Details

Assessment title	Abr.	Weighting
Project Assessment	WRIT1	100%

Pass marks are 40% for undergraduate work and 50% for postgraduate work unless stated otherwise.

Task/assessment brief:

There are two parts to the assessment:

- **Part I: Smart city project (50%)**
- **Part II: System Diagrams (50%)**

Part I: Modern City project

This project should be implemented in Java and using a MySql server/MS Access database. This is a web-based project that stores all the important details of the city. Cities and urban areas are experiencing a large influx of individuals seeking jobs, education, and perhaps a better way of life. However, in the initial days after the move, people are unfamiliar with the city's main facilities, attractions, and services. The smart city initiative (https://commission.europa.eu/eu-regional-and-urban-development/topics/cities-and-urban-development/city-initiatives/smart-cities_en) aims to address this by developing an integrated platform for storing vital and related information that will assist newcomers in navigating a city.
By using this project, visitors, students, and job seekers will be able to access information such as hotels, rental facilities, transportation services, healthcare services, airline ticket reservations, shopping sites, emergency helpline, and virtually everything else they would need when visiting a new city. It's equivalent to a smart city guide for new visitors.
Users can access the application through the Internet and search all the smart city web pages for the information they require. For example, users may see a city's map, as well as its most prominent landmarks. This information will make navigating the city a lot easier for them.
The smart city project has five modules:

1. **Administration**

   It is the application's central controller. It oversees the upkeep of the other four modules and uploads all new information to the site. It also authenticates user-profiles and uploads all further information to the site.

2. **Tourism**

   As its name implies, this module is responsible for all tourism-related operations in the city, including hotels, restaurants, tourist attractions, ATMs, theatres, and so on. A user authenticated by the administration module becomes the primary user of this module.

3. **Student**

   This module helps students move around the city. It provides students with all academic information, such as the locations of the outstanding educational institutes, libraries, coaching centers, colleges, universities, and so on.

4. **Job applicant**

This module contains key information on the city's job opportunities in different departments. In addition, users have access to a wide range of job-related data from a variety of industries. The main goal of this module is to help the city administration struggle with unemployment issues.

5. **Business**

   This module focuses on providing news, information, and opportunities in the city connected to the business. People can obtain information on the city's business centres and industries.

Project Description

1. **Designing the Window**

   Create the project window and add a background image. Make sure to keep the image in the same directory as the project to avoid inconsistencies. Next, your code must set the title of the library project window with an appropriate library name. Then, when you run the code, the window should appear. Explain the process step by step.

2. **Connecting to the Database**
   You have to connect the server with the correct credentials associated with the MySql server/MS Access installed on our system.

3. **Adding the Buttons/links**
   Add the buttons to the window frames. Then, write the code and explanation. It is also required to connect each button to the database.
   - Delete record button
   - Update record button
   - Search record button
   - Add record button

Part II: System Diagrams

Draw UML diagrams (Use cases and Class diagram) and Report for the project

- Class diagrams
- Use case diagrams
- Sequence diagrams

Submitting your project

You are required to submit the software components (source code), and relevant documents that contain the following

- **Source files** zip the java project and upload
- **Documentation (Word/PDF document)** satisfying the requirements identified.

Word count (or equivalent): 4000 words

This is a reflection of the effort required for the assessment per student (team member). Word counts will include any text, tables, subtitles and citations. Reference lists are excluded from the word count.

Academic or technical terms explained:

Key Bloom elements:

Comprehension – Demonstrate key aspects needed for the programming and design.
Analysis – Comparing and contrasting different methods of interaction and use cases.
Synthesis – Building awareness of industry standards.

Submission Details

Submission Deadline:	8th of December, 2023.	**Estimated Feedback Return Date**	This will normally be 20 working days after initial submission.
Submission Time:	By 4.00pm on the deadline day.		

Moodle/Turnitin:

Any assessments submitted after the deadline will not be marked and will be recorded as a non-attempt unless you have had an extension request agreed or have approved mitigating circumstances. See the School Moodle pages for more information on extensions and mitigating circumstances.

File Format:

The assessment must be submitted through the Turnitin submission point in Moodle. You can use any support file type for your report (Word, PDF for example)

Your assessment should be titled with your:

student ID number, module code and assessment ID,
e.g. st12345678 CIS6003 WRIT1

Feedback

Feedback for the assessment will be provided electronically via Moodle. Feedback will be provided with comments on your strengths and the areas which you can improve. View the guidance on how to access your feedback.

All marks are provisional and are subject to quality assurance processes and confirmation at the programme Examination Board.

Assessment Criteria

Learning outcomes assessed

Learning Outcomes
[LO1] Demonstrate fluency in contemporary programming languages, development tools and environments.
[LO2] Evaluate and demonstrate the theory and concepts of contemporary/industry standard programming and design in the software development life cycle.
[LO3] Demonstrate awareness of industry standards of professional and ethical software development, software carpentry and codemanship.

Assessment Criteria

Developer Documentation	**100%**
Part I Assessment - (**LO1, LO2, LO3)**	50%
Part II Assessment - (**LO1, LO2)**	50%

Other skills/attributes developed

This includes elements of the Cardiff Met EDGE (Ethical, Digital, Global and Entrepreneurial skills) and other attributes developed in students through the completion of the module and assessment. These will also be highlighted in the module guidance, which should be read by all students completing the module. Assessments are not just a way of auditing student knowledge. They are a process which provides additional learning and development through the preparation for and completion of the assessment.

Ethical	Usability, and user experience are two important factors when designing and implementing a software application that is interacted by the users. The information, messages generated by the system, and information collected by the client program (e.g., personal data) should be handled, saved and processed with proper care adhering to international and regional laws and regulations.
Digital	Software engineering and programming skills accumulated during the course and through the assignment will enhance student's hands on digital skills, use of IDEs, build tools, inter-component/process communication skills, and industry standard programming and design best practices.
Global	Software Engineering principles design patterns discussed in this class universally accepted. Skills gained through this are location and language agnostic and are applicable across the globe.
Entrepreneurial	Problem solving skills, knowledge acquired on client-server architecture, programming and

	system designing skills are pivotal for anyone who expects to aspire to be an entrepreneur in a technical domain.

Marking/Assessment Criteria

Part I	**50%**
Administration	20%
Tourism	20%
Student	20%
Job Applicant	20%
Business	20%
Total	100%

Part II	**50%**
Class diagrams	10%
Use case diagrams	10%
Sequence diagrams	10%
Coding skills and best practices, appropriate use of functions, variables, loops, and arrays	40%
Documentation explaining the implementation, execution details	20%
Overall execution of the system	10%
Total	100%

Mark Classification	Criteria
70 – 100% (1st)	An excellent design and implementation is given that covers all the requirements of the assignment. Outstanding coding practices are followed, git repository is well maintained with continuous commits and comments. Exceptional use of multithreading, exception handling (including custom exceptions), and excellent demonstration of Java OO programming concepts (nested classes, interfaces, collections, Lambdas etc) and design patterns. Additional elements have been considered to improve usability and user experience of the system. Excellent, detailed documentation is provided explaining the build, and execution of the program.
60-69% (2:1)	A good design and implementation is given that covers most of the system requirements. Good coding practices are followed, git repository is maintained with some commits and comments. Good use of multithreading, exception handling, good demonstration of Java OO programming concepts, and knowledge in design patterns. A good documentation is provided explaining the build, and execution of the program, and experimental results.
50-59% (2:2)	An average design and implementation is given covering only some of the system requirements. Demonstrate average coding skills and git repository is not well maintained. Some of use of threads, exception handling and design patterns is visible. An average documentation is provided explaining the build, execution of the program, and experimental results.
40-49% (3rd)	A marginal design and implementation is given covering only few system requirements. Demonstrate basic coding skills and git repository is not well maintained. Basic (or no) use of threads and exception handling is visible and no use of design patterns. Basic documentation is provided explaining

	the build, execution of the program, and experimental results.
<40% (Fail)	A poor design and implementation is given covering only very basic system requirements. Demonstrate poor coding skills and git repository is not well maintained (or not provided). Poor (or no) use of threads and exception handling is visible. Very basic documentation is provided explaining the build, and execution details of the program. No meaningful design has been given and little to no code has been implemented. There is no evidence of any meaningful testing or analysis of the results obtained. Little to no understanding of the problem and the techniques required to create a solution are evident

Further Information on assessment, referencing and grading can be found in the Module Handbook (or Moodle)

Cardiff Met

MetCaerdydd