**Kingston University, BSc (Hons) (top-up)**

**Coursework Coversheet**

**Draft Coursework – Subject to Moderation**

**Part 1 - To Remain with the Assignment after Marking**

|  |  |
| --- | --- |
| **Student ID:** | **Student Name:** |
| **Module Code:** | **Module Name:** |
| **Assignment number:** | **ESoft Module Leader:** |
| **Date set:** | **Date due: 21st March 2025** |

**Guidelines for the Submission of Coursework**

1. Print this coversheet and securely attach both pages to your assignment. You can help us ensure work is marked more quickly by submitting at the specified location for your module. You are advised to keep a copy of every assignment.

2. Coursework deadlines are strictly enforced by the University.

3. You should not leave the handing in of work until the last minute. Once an assignment has been submitted it cannot be submitted again.

**Academic Misconduct**: **Plagiarism** and/or **collusion** constitute **academic misconduct** under the University's Academic Regulations. Examples of academic misconduct in coursework: making available your work to other students; presenting work produced in collaboration with other students as your own (unless an explicit assessment requirement); submitting work, taken from sources that are not properly referenced, as your own. By printing and submitting this coversheet with your coursework you are confirming that the work is your own.

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | ESoft Office Use Only:  Date stamp: work received | | |  | | --- | |  | |

**Kingston University, BSc (Hons) (top-up)**

**Coursework Coversheet**

**Part 2 – Student Feedback**

|  |  |
| --- | --- |
| **Student ID:** | **Student Name:** |
| **Module Code:** | **Module Name:** |
| **Assignment number:** | **ESoft Module Leader:** |
| **Date set:** | **Date due:** |

|  |
| --- |
| Strengths (areas with well-developed answers) |

|  |
| --- |
| Weaknesses (areas with room for improvement) |

|  |
| --- |
| Additional Comments |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **ESoft Module Lecturer:** | **Provisional mark as %:** |  |
| **ESoft Module Marker:** | **Date marked:** |

**Kingston University, BSc (Hons) (Top-up)**

**CI6125 Software Development Practice – 2025**

**Esoft Module Leader:**

**Submission deadline**

**Coursework 1 & 2:**

**Software Development Practice Coursework 1&2 (Assessment element 1 - Group coursework and Assessment element 2 - Individual coursework / artefacts)**

**Overall system requirements:**

You are required to produce a solution using appropriate process, methodology to analyse, design, implement and test the software for the scenario described below:

**Scenario**

Your company has been assigned of developing a Smart Campus Management System for a university. Through an integrated system, the platform aims to manage campus events, maximize resource allocation, and enhance staff and student participation. The goal is to promote a cooperative and productive campus culture while streamlining operations.

The intended system must accomplish the following high-level objectives:

* To allow academic staff, administrative personnel, and students access to campus services such as scheduling, announcements, and resource reservations through a unified, integrated platform.
* To provide a platform for scheduling classes, workshops, and events, as well as for managing extracurricular and academic activities.
* To encourage collaboration amongst academic staff, administrative personnel, and students by providing resources for communication, group interactions, and resource sharing.
* To maximize campus resource consumption, including equipment allocation and event planning.
* To generate real-time analytics and statistics on resource utilization, event attendance, and overall campus involvement.

It is also noted that the intended software should facilitate the following features.

* Administrators, lecturers and students can all have role-based accounts. Users can register their accounts log in and create profiles. Users will find it subsequently easier to view management-based items and tasks and to interact with role-based items.
* Provide students with a flexible and dynamic scheduling tool to manage their academic calendars so they can view their schedules, register for classes and receive notifications of any modification.
* Provide the option for users to reserve equipment classrooms and other facilities online. It will also facilitate scheduling conflict avoidance and resource availability monitoring.
* The site provides tools for planning announcing and managing events - including seminars guest lecturers and student councils.
* Provide administrator dashboards to monitor campus events in real time and view detailed information about attendance resource usage and events.
* Make use of file sharing task management group messaging and task sharing to facilitate communication between lecturers and students and to promote group projects and discussions.
* Allow users to personalize their email, SMS, or in-app alerts, and inform students of upcoming events, administrative changes, or class schedule adjustments.
* To protect sensitive user data, encryption, secure access restrictions, and compliance with institutional and data security regulations are all essential.

1. The system should have two components.
   1. Back-end server application with appropriate persistence storage.
   2. Front-end client application with responsive features.
2. Implement single registration per user identity with robust security measures to prevent duplicate accounts.
3. All users and their levels should be within the administrators' control.
4. When implementing the software system, should consider security and privacy measures, mobile responsiveness and cross-platform compatibility features.

By considering the brief outline requirements given above, you should try to derive and justify the hidden and implicit requirements.

THIS COURSEWORK HAS TO BE COMPLETED IN GROUPS OF FOUR TO FIVE STUDENTS. Please form the groups within your batch only.

Note: *The security mechanisms should be considered as Top priority in this application system.*

**Deliverables**

There are two major submissions.

1. A group submission covering the overall product design and development done at group level (Coursework 1)
2. An individual submission covering component level work carried out by you as a member of the team (Coursework 2).

**The final submission on Kingston Canvas should consist of a zipped folder containing the required documentation and relevant resources, such as software, codes, libraries and etc. The documentation must include two main sections: Coursework 1, which is the group report, and Coursework 2, which is the individual report. Both sections should be included in a single Word document.**

1. **Coursework 1 - Group submission (30% of the module mark)**

Supply professional standard product documentation presenting the evidence for completing the following aspects of the system development tasks and the associated artefacts.

1. Introduction and background – introduce the overarching need for the system, and your overall approach.
2. Software requirements specification, including discussions on system analysis tasks and their outcomes.
3. Software design including the system architecture design, and the system specification. Here you should present how individual components and services have been designed to meet the underlying requirements of the system. There should be sufficient discussion on the separation of concerns, how component communications are taking place, and security concerns you have considered.
4. Implementation – development of the system using appropriate programming language, tools, frameworks etc.
5. Software quality approaches adopted, including testing strategies, validation and verification approaches and evaluate their effectiveness in producing quality software.
6. Clarification on the use of software tools for the project implementation including collaboration tools. Each group should explain the use of software development tools and collaboration tools. In this case, Cloud tools are to be used such as Git, Bitbucket, JIRA, selenium, etc., based on the purposes.

The report must prominently display the list of team members and their respective contributions, expressed as percentages, for each of the six tasks that were collectively agreed upon through consensus. The overall group mark will be distributed to individual group members according to their individual contributions to the tasks. If this detailed information is not provided, a statement should be included, indicating that all team members contributed equally. In such a case, all team members will receive the same group mark.

1. **Coursework 2 - Individual submission (30% of the module mark)**

Each member of the group must implement a part (two or more components) of the software. The components you are responsible for should be clearly indicated, referring to the overall architecture diagram (provided as part of task 3 of the group work).

The individual report submission should cover the below.

1. Introduction to the components you were responsible for, functional features covered by each of those components, their scope, and boundaries, and how they communicate within the system.
2. Presentation of each component with suitable diagrams, discussions etc. covering its interface, workflow design, and development. This should contain the appropriate code snippets, and clear indication on the underlying workflow, any algorithms involved.
3. Test plan and test outcomes for the respective components. Each member of the group must test your implemented software components, while following the overall testing strategy agreed at the team level.

**Marking criteria**

**Group submission (30% of the module mark)**

|  |  |
| --- | --- |
| **Element** | **Marks available** |
| 1. **System introduction and background –** A clear description of the problem domain being tackled, clear indication on the problem understanding, and the overall approach taken by the team to understand the problem at holistic, and functional levels. Evidence of additional research. | **15** |
| 1. **Software requirements specification** – Clear list of functional and non-functional requirements, with suitable discussions on the requirement elicitation tasks. | **15** |
| 1. **Software design** – System design process, and design models including the system architecture; architecture diagram, interface models, UML models etc**.** | **15** |
| 1. **Implementation –** Evidence for working prototype or built system versions, with justification for the selected methodology. | **30** |
| 1. **Software quality assurance -** Software-test design, test implementation and test report, evidence of the testing with prepared test data as specified in the requirements specification, justification for the selected methodology. Acceptance testing is a must. | **15** |
| 1. **Use of collaborative tools –** Properevidence for the use of appropriate collaborative tools and automating tools**.** | **10** |
| **Total** | **100** |

**Individual submission (30% of the module mark)**

|  |  |
| --- | --- |
| **Element** | **Marks available** |
| 1. **Introduction to components –** Presentation of the components covered by the individual work, referencing to the architecture diagram of the system. Clear description of the functional requirements or services covered by respective components, scope of the service, component boundaries, and communication mechanisms in place. Appropriate focus on cohesion and coupling aspects. User stories covered should be elaborated. | **10** |
| 1. **Component presentations** – Comprehensive introduction to components, clear indication of the functions covered, their dependencies with other components, design of the components work-flow, interfacing, data flow etc. Suitable discussion on the development steps, challenges faced, solutions adopted, evidence for interactions. Limitations or future work associated with any failed or incomplete components. Preparation of cost estimation for the individual component with the approach. | **50** |
| 1. **Component level quality assurance –** Test plan and test outcomes for each of the components, with clear evidence (e.g. screen captures, log files). Evidence for Test driven development and/or test automation where appropriate. Discussions on other quality assurance methods adopted/recommended. | **20** |
| 1. **Use of collaborative tools & Implementation –** Evidence for working prototype or built system versions, with justification for the selected methodology and use of appropriate collaborative tools and automating tools. | **20** |
| **Total** | **100** |

**Academic Integrity:**

Academic integrity means demonstrating honest, moral behaviours when producing academic work. This involves acknowledging the work of others, giving appropriate credit to others where their ideas are presented as part of your work and the importance of producing work in your own voice. Contributions by artificial intelligence (AI) tools must be properly acknowledged. As part of a learning community students share ideas and develop new ones - you need to be able to interpret and present other people's ideas and combine these with your own when producing work.

**Plagiarism (including copying, self-plagiarism and collusion)**

The act of presenting the work of another person (or people) and/or content generated by artificial intelligence (AI) tools as your own without proper acknowledgement. This includes copying the work of another student or other students.

The University expects students to take responsibility for the security of their work (i.e. with written work, to ensure that other students do not get access to electronic or hard copy of the work). Failure to keep work secure may allow others to cheat and could result in an allegation of academic misconduct for students whose work have been copied, particularly if the origin of the work is in doubt.

**Self-plagiarism**

The act of presenting part or all of your work that has been previously submitted to meet the requirements of a different assessment, except where the nature of the assessment makes this permissible.

**Collusion**

The act, by two or more students of presenting a piece of work jointly without acknowledging the collaboration. This could include permitting or assisting another to present work that has been copied or paraphrased from your own work.

The University also defines collusion as the act of one student presenting a piece of work as their own independent work when the work was undertaken by a group. With group work, where individual members submit parts of the total assignment, each member of a group must take responsibility for checking the legitimacy of the work submitted in his/her name. If even part of the work is found to contain academic misconduct, penalties will normally be imposed on all group members equally.

**Purchasing or Commissioning**

The act of attempting to purchase or purchasing work for an assessment including, for example from the internet, or attempting to commission, or commissioning someone else to complete an assessment on your behalf.

The procedures for investigating suspected cases of academic misconduct are set out in Academic Regulations 6 Academic Integrity - Taught Courses 2023/24

**Acknowledging Generative AI in coursework**

Where generative AI has contributed to an assignment the following information should be included in the submission:

A statement on the use of generative AI as part of the assessment, including the extent of use, and how it was used as part of all stages in creating the final submission, e.g., including planning, and generating ideas. This should normally be provided at the end of a written assignment with the heading ‘Acknowledgement of AI Contribution’. For other assignment types, module staff will advise on how this should be done.

**You must meet all deadlines set. Failure to do so will result in a penalty.**

Work submitted late but within a week of the deadline will be capped at 40% and receive a grade of LP (Late Pass) unless it is not of a passing standard in which case it will receive a grade of LF (Late Fail). Work submitted beyond a week of the deadline without approval will get 0% with a grade of F0. If, however, you have a serious problem, which prevents you from, meeting the deadline you may be able to negotiate an extension in advance. In the first instance you should contact the module team for advice. However, any extension will need to be formally agreed by the Faculty via the Mitigating Circumstances process, your work will then be marked without penalty.