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

**Draft Coursework – Subject to Moderation**

**Coursework Cover Sheet**

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

|  |  |
| --- | --- |
| **Student ID:** | **Student Name:** |
| **Module Code:** | **Module Name:** |
| **Assignment number:** | **ESoft Module Leader: Ms. Lasanthika Jayasekara** |
| **Date set:** | **Date due: 14th of April 2024** |

**Guidelines for the Submission of Coursework**

1. Print this cover sheet 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 Cover Sheet**

**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 – 2023/24**

**Submission deadline: 14th of April 2024– before 3.55pm via ELMS**

**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 software firm was assigned to develop a software system for a community-driven volunteering platform. With the help of the portal, volunteers will be able to find purposeful opportunities, register for initiatives, and keep track of their contributions all in one place. The objective is to develop an approachable platform that enables people to positively influence their communities and beyond.

The intended system needs to be able to accomplish the following significant high-level objectives:

* To facilitate individuals to identify and engage in volunteer opportunities that match their interests, abilities, and availability; also, to give volunteers a platform to interact with nonprofitable organizations, community groups and initiatives that require assistance.
* To identify, organize, and manage volunteers for their programs and events in order to enable them to give back to their communities in a meaningful way by using their time, talents, and resources to solve social and environmental issues.
* To work collaboratively and build a network of volunteers, businesses, organizations, and other stakeholders to address community problems and bring about constructive change.
* To make volunteering more enjoyable and fulfilling for all volunteers by supplying tools to assist with planning, communicating, training, acknowledging accomplishments, and providing feedback.
* To monitor and report on the overall impact of organizations and volunteers, including data on hours of service, completed projects, beneficiaries contacted, and social outcomes attained.

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

* User signup, login, and profile creation, where users can also input their information, interests, abilities, and talents. Profiles should include options for individual volunteers and organizations in need of volunteers.
* A searchable database of volunteer opportunities, organized by type, (one-time, long-term, ongoing), location (city and region), objectives (teaching, cleaning, building etc.), or time based. Users should be able to filter and sort possibilities by their level of expertise, availability, and interest in the system.
* Allow users to sign-up for volunteer roles directly through the platform, entering specific details such as times they are available, contact information, qualifications and any background checks or credentials that are necessary. After registration process, notify the volunteers via email to confirm acceptance of their volunteer project registration.
* Enable volunteers to monitor their hours, activities, and overall contributions over time by tracking the volunteer hours and generate relevant reports.
* Provide messaging tools including group chats and direct messages to facilitate communication between volunteers and groups.
* Enable volunteer organizations and groups to evaluate, discuss, or rate one another based on their shared volunteer experiences.
* Make an easily accessible schedule of events with information about forthcoming training sessions, volunteer opportunities, updates, and community gatherings.
* Include robust security measures in place that adhere to privacy laws and data protection guidelines, safeguard confidential information, and prevent unauthorized access.

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. The volunteers should be able to register in the system and multiple registrations per identity should be prevented.
3. All users and their levels should be within the administrators' control.
4. Provide a searchable database of volunteer opportunities categorized by location, cause, type of work and time commitment.
5. Should be able to track volunteer hours and generate relevant reports.
6. Features such as communication and messaging, feedback and ratings, event calendar reminders, and notifications should be implemented in the software system.
7. 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 (assessment element 1)
2. An individual submission covering component level work carried out by you as a member of the team (assessment element 2).
3. **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 primary 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 final submission to Kingston Canvas should be a group submission, presented as a group report along with the necessary resources (such as software code and libraries) provided in a zipped folder. This submission should be made by the team's designated manager, which means there should be one submission per group. 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 submission should include a report, associated software elements (code, libraries etc.).

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 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 –** good evidence 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** |

**Level of work expected:**

This is a major piece of work, and it is expected that you will need to do some very thorough research and that ideally your research will be as up to date as possible given that this is a very rapidly moving field. Work containing vague descriptions or unsupported assertions will be penalised.

**Feedback:**

You can invite the module-staff to review your progress and provide formative-feedback.

**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 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

**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.