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

**CI6125 Software Development Practice – 2023/24**

**Esoft Module Leader: Dr Lochandaka Ranatunga**

**Submission deadline: 21st of December 2023 – before 3.55pm via ELMS and before 3:55pm via Canvas**

**Software Development Practice Coursework (assessment element 1 - Group coursework and assessment element 2 - Individual 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 to facilitate general public to make complaints against environmental crime incidents through online and mobile based applications with evidence directly to two institutions, namely Wildlife conservations and Forest conservations. The two institutions should be able utilise these complaints to take preventive, protective and legal actions for the conservation of wildlife, forestry and environmental resources. Further the system should provide necessary facilities for incident progress monitoring and reporting for key stakeholders. The intended system should be able to satisfy the following key top-level objectives:

* To enable public to submit complaints against wildlife, forestry and environmental crime activities through the application system using online web system and/or mobile application.
* To take wildlife conservation and forestry conservations institutes under a one network of communication and complaints management.
* To facilitate each institution under the system to manage the procedural workflow of investigations triggered through a complaint directed to them and report the status of the investigation with greater accountability with respective to their responsibilities.
* To provide divisional, institutional, national level dashboards for wildlife, forestry and environmental crime incident monitoring, reporting and decision support mechanism.

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

* Facilitate with multiple user roles and their privileges within the system to match with the user levels available in each institutional framework while providing the facilities for the user role and privilege customisation capabilities.
* Software should be able to capture the actions taken by each responsible officer related to directed complaints with evidence of the investigations. This includes the reporting of progress of incident investigations with locations, photos, time frame etc.
* Software should facilitate the incident management workflow appropriately (logically) for each institution into the system with their divisions, branches, beat offices and different roles and officers etc.
* Software should be designed to empower necessary security measures to provide officers’ mobile applications with relevant device-based authentication.
* There should be a disaster recovery mechanism for the entire system to keep up with high availability.
* The availability of audit trail for all actions on the system where the highest-level system administrator should be able to review the records when needed.
* The officers should be able to indicate the actions that have been taken and update the status complainer to be kept informed all times. **You are required to design a web and mobile apps to achieve above objectives.** High-level system component requirement are as follows.

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 complainers should be able to register in the system and multiple registrations per identity should be prevented.
3. Administrators should be able to manage all users and their levels.
4. Field / beat officers should be able to mark the status of the reservation upon customer activities accordingly.
5. The complaints should be directed to the relevant area office and the institutions based on the characteristics of the complaint automatically as much as possible, otherwise, they should be allocated manually.

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; a group submission covering the overall product design and development done at group level (assessment element 1), and an individual submission covering component level work carried out by you as a member of the team (assessment element 2).

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 demonstrate the use of software development tools and collaboration tools during the demonstration, which is a key component. In this case, Cloud tools are to be used such as Git, Bitbucket, JIRA, selenium, etc., based on the purposes.
7. Product presentation covering the system design, development and evaluation processes and the product demonstration. A video of the working product should also be provided.

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

Each member of the group must implement a part (two or more components) of the software. i.e. functions such as the login, complaint recording, incident monitoring, daily report service and so on. 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.) and **an individual presentation**.

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.

For example, if you were responsible for two components, your individual report should contain the overall introduction (element 1 above), component-1 presentation, component-1 testing, component-2 presentation, component-2 testing (i.e., elements 2 and 3 should be repeated for each component).

1. An individual presentation demonstrating the components you were responsible for. Here, you need to also present the approach and tools you adopted to complete the allocated tasks, how you collaborated at team level, any challenges faced, outcomes and your reflections on the overall experience.

**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. | **10** |
| 1. **Software requirements specification** – clear list of functional and non-functional requirements, with suitable discussions on the requirement elicitation tasks. | **10** |
| 1. **Software design** – system design process, and design models including the system architecture; architecture diagram, interface models, DFD, 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** |
| 1. **Group presentation –** group’s presentation of the overall system development process, system demonstration, system evaluation outcomes, and overall reflections. | **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. **Individual presentation –** Comprehensivepresentation of the component development process. Demonstration of software development and quality assurance skills and awareness. Evidence for good collaborative work practices, use of appropriate tools. Ability to answer questions on the tools, process followed, implementation outcomes and future work. | 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 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

**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.