|  |  |
| --- | --- |
| **COURSEWORK ASSIGNMENT** | University of East Anglia  School of Computing Sciences |

**MODULE: CMP-5012B**

**ASSIGNMENT TITLE: Stage 2: Web-based implementation**

**DATE SET : 20th of March 2019**

**DATE & TIME OF SUBMISSION : 15th of May 2019, 3PM**

**RETURN DATE : 12th of June 2019**

**ASSIGNMENT VALUE : 60% (10% demo, 50% report+code)**

**SET BY : Rudy Lapeer**

**CHECKED BY : Debbie Taylor**

|  |
| --- |
| **Aims**   * Design of a software system of interrelated entities; * Undertake business logic analysis and design for an extensive web-based software engineering project; * Understand the architectural and design principles underpinning robust software systems; * Recognise and formalise application logic, and update design and implementation accordingly.   **Learning outcomes**   * Experience in designing a component-based, extensible and scalable software application; * Improved software development skills; * Ability and experience of working in a team developing an industrial-grade software application;   **Assessment criteria**   * Systematic and consistent analysis of application logic; * Consistent diagrams documenting application logic and the system design for the software architecture; * Identification and suitable design of classes and components; * Implementation and testing; * Demonstration; * Documentation of Software Components and Effective Management Report Writing. |

|  |
| --- |
| **Description of assignment**  Now that you have gone through at least one iteration of the software engineering cycle and developed a desktop-based application with own-designed MVC architecture, the next stage involves further iterations to arrive at a web-based implementation of the same application. You are allowed to use a framework that includes a built-in MVC (or similar) pattern (suggested frameworks are Django and ASP.net).  Your final deliverables should include:   * Updated analysis of similar software systems with a clear feature matrix (if changed); * The next items require updating after every iteration:   + User stories, associated feature descriptions and use case descriptions;   + OOD: UML class diagrams; integration in platform’s MVC (or equivalent).   + Sequence diagrams, state diagrams;   + Testing (unit and integration). * Final software product: web-based application with front- and back-end and database (full stack).   **Required:**   * **A report;** * **System implementation** (Delivered via Git through bitbucket); * **Demo.** |

|  |  |
| --- | --- |
| **COURSEWORK MARKING SHEET** | University of East Anglia  School of Computing Sciences |

**Module: CMP-5012B**

**Stage 2: Web-based implementation**

**To be completed by the student:**

|  |
| --- |
| **Registration No:** |

**We certify that this work is our own, undertaken wholly for this assignment, and has not been copied in whole or part from other sources, except where specifically acknowledged.**

**Date:**

**To be completed by the marker.**

**Assessment criteria are specified below:**

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Demonstration of your software system.\* | 100% |  |  |
| 2. Report – document:  Use case and feature descriptions, user stories;  Non-functional requirements;  Architectural (MVC), class, sequence and state diagrams;  Evidence of unit and integration testing;  Evidence of iterative and incremental development (IID). | 40% |  |  |
| 3. Final implementation of software system (including code documentation).\*\* | 60% |  |  |

\* (1) total module weight = 10%, \*\*(2)+(3) total module weight = 50%

**Provisional Mark Awarded: % Date:**

**Marked by: Checked by:**

**Please remember that students should retain all returned coursework and have it available for External Examiners, if required. All marks are provisional until confirmed by the Board of Examiners.**

**FURTHER INFORMATION ON STAGE 2 IMPLEMENTATION AND REPORT**

The Stage 2 report adopts the same format as the Stage 1 report. It may differ as follows:

* Provide the first iteration in detail just like in your first report.
* For any subsequent iteration: provide **updates** on user stories, UML diagrams, non-functional requirements and testing.
* Test (unit and integration) have to be performed and reported per iteration (only updates for every next iteration).
* Non-functional requirements (NFRs) need to be considered – check the lecture notes on software quality to help you which NFRs suit your project. Examples are:

Usability (Clear and intuitive UI), Security (password/data encryption), Validity (i.e. user input validation), Robustness (see later), Clean Code (remove code smells), …

**OBSERVATIONS FROM STAGE 1 TO BE CONSIDERED (AND DEMO-ED) IN STAGE 2**

* Some Stage 1 developments had poor (incomplete, nonsensical, corrupted) application logic. Application logic needs to make sense and depends on your application (health tracker, study planner, rental/booking website).
* This logic needs to be thoroughly (user) tested in Stage 2 to ensure that the application is capable of doing a full cycle of user interaction (which may span weeks or months depending on the application) without falling over (e.g. crashing, lost or incorrect data display). This will have to be demonstrated during the demo assessment.
  + For example, in the health tracker you have to show that someone can enter weight, diets over a period of time (with correct data retention) and that this can be updated and plotted anytime when asked.
  + In the study planner, you have to be able to show a history of previous, pending and non-completed activities at any instance in time.
  + In rental/booking systems you have to be able to show the current rentals, bookings and idle resources at any instance of time and update it with new rentals, bookings, returns anytime.
* The above implies that you need to populate your database with data of a sufficiently large user cohort over a sufficiently large period of time (weeks, months). One user with a few activities (as seen in Stage 1) is insufficient to test the **robustness** of the system.