VANIER CÉGEP/COLLEGE | Continuing Education

Software Capstone Project: Internship Alternative

Context and Description

Given the current COVID context, where placement of students is difficult for their internships, we understand that some employers do not have the means of managing interns. Continuing Education is therefore offering the students the alternative option of completing their internships through a Capstone project. This project allows students to build real-world experience while also fulfilling the requirement that would otherwise be satisfied in the Software Applications Specialist internships.

We wish to provide students with a meaningful experience that can substitute that of an Industry internship and satisfy the competency statements for this portion of the program.

Time Commitment

Projects require the same amount of work as an internship, and the internship coordinator acts as your project supervisor. If a student chooses this option, they must start **as soon as their week after their last day of class in their program**. Projects are a duration of **9 weeks and 300 hours**. In the context of COVID-19, students will work from home.

Role of the Internship Coordinator

As a substitute to internal supervision, students could regularly contact their internship coordinator to help orient the students in their tasks. The goal being to help the students develop professionally, the coordinator does not have the role of a teacher and does not teach technical skills to the students, but merely orients the student toward references they can use in furthering their development.

Proposed Project & Evaluation (choose only one)

1. Development of a Major Web Application

The student will act as product owners for a Web application that they will develop.

- Including approximately 30 user stories for a student.
- Developed on a major framework (e. g. Laravel, ASP.net, Spring)

- Complete with RESTful API.
- With a complete automated testing suite (e. g. Unit Test, Integration Test)
- With database designed by the students. (e. g. SQL server, MySQL, Oracle)
- Applying internationalization (i18n) techniques. (Plus)
- Perform input validation.
- Graphical User Interface with good Human Machine Interaction interface.
- Mobile compatible and mobile optimized (plus).
- Basic documentation in a README file
- Well-designed icons and graphics
- Social Media Sharing

2. Industry-level Desktop Management Information System

The student will design and implement an MIS application that solves a real-world problem. MIS is used for decision-making, and for the coordination, control, analysis, and visualization of information in an organization.

- Students can use Java, C++, C#, Python, or Visual Basic as the main programming languages.
- Examples include Library Management System, Accounting System, Clinic Management System, and others.
- Student should use Object-oriented design.
- Perform input validation.
- Perform control access rules.
- Including approximately 30 user stories for a student.
- Graphical User Interface with good Human Machine Interaction interface.
- Use well-stablished software engineering design patterns for the chosen platform
- With a complete automated testing suite (e. g. Unit Test, Integration Test)
- Basic documentation in a README file.
- Version control system, project management and issue tracking will be used throughout the whole development process.
- Applying internationalization (i18n) techniques. (Plus)
- With database designed by the students.
- Setting and configuration

3. Development of an Industry-level Mobile Application or Mobile compatible Website

The student will act as product owner for a Mobile application/website that solves a real-world problem. The App will be developed in Android Studio, Xamarin, PHP, Java, ASP or some other language/framework. Version control system, project management and issue tracking will be used throughout the whole development process.

Purpose: this is an entry-level mobile project for students entering the field. Students will design, implement, test and publish their product.

The development process and final product will provide the following:

- Design of UI schemas using a mock-up tool
- Multiple screens (5 at the very minimum)
- Re-use of screen components (e.g. re-use of Fragments in Android Master/Detail flow)
- Data communication between screens, or components of screens
- A responsive interface, which adapts to different screen sizes and orientation
- At least 3 types of menus (drawer, top bar, contextual, etc.)
- Well-designed icons and graphics
- App notifications
- App settings
- Applying internationalization (i18n) techniques. (Plus)
- Implementation of an external database (local or remote) including at least 3 tables
- Consume information from an online RESTful API
- A responsive, which performs long tasks in the background
- Use well-stablished software engineering design patterns for the chosen platform
- A complete automated testing suite
- Basic documentation in a README file
- A LICENSE file
- Publication the application in an App Store in case of application
- Social Media Sharing

Project Development Phases:

The student will develop the project in phases as follows:

- Project Proposal:
    - Students must submit a proposal and do a short presentation of their ideas before the end of the third week. The idea must be accepted.
- Project Planning/Initialization and Analysis

  - Assess the size, scope, and complexity of the project, and establishes procedures to support subsequent activities.
  - Analysts identify detailed system requirements
  - Requirements capture and modelling undertaken
  - Information stored within a CASE tool includes DFD, UML, etc.
  - Writing user stories
- Project Design and Implementation
  - Developers work from design specification to produce the software
  - Database table, and design
  - Interface design
  - Coding
- Project Testing
  - Writing unit tests that covers the defined user stories, and the code.
  - Integration and Acceptance Tests
- Project Documentation and Presentation
  - Manual/User documentation is provided to ensure that the system users have enough clear information to enable them to make use of the system
  - Standard Software Design Document explains how a software product, or a feature are built to meet a set of technical requirements.
  - 15-20-minute Project presentation using Power Point or any similar software.

Evaluation:

Criteria [1]
Project Proposal (Planning/Initiating phase) Think about the internship coordinator as a potential buyer of your project • Is it a new idea (never been implemented before)? • The presentation slides, coverage and presentation skills • Potential implementation and release in reality
Project Analysis Think about the internship coordinator as your project manager • How detailed the analysis is (does it cover all requirements) • Diagrams (DFD, UML, State-machine, etc.)
Design and Implementation Think about the internship coordinator as a code reviewer in your company • Database Design • Code correctness percentage (Does it work?) • Time and space complexity of the algorithms used in the code • Recommended code practices • Good Human Machine Interaction (HMI) practices
Testing

Think about the internship coordinator as the QA manager in your company • Unit test coverage • Integration and Acceptance Testing
Documentation Think about the internship coordinator as a potential user/developer of the project • User Manual • Software Design Document (SDD)

[1] The requirements of each project explained in page 1 to 3 will be considered at each phase of the evaluation.