**Student Guides Document**

**Application Development Project (SWP391) Topic**

**------- \* -------**

# I. Introduction

The purpose of this topic is for students to practice on developing the software application in working teams (project-based) following the industry-like software development activities of analyzing, designing, and coding. The success project team has to deliver the required deliverable items on time for the sufficient working scope with acceptable quality.

The project lifecycle is divided into 3 phases

* ***Initiation***: students to receive team & work assignments, study and propose solutions to complete the assignment. Submit & explain the work results to the teacher for evaluating. This phase lasts for two weeks.
* ***Construction***: build the software package and prepare the documentation as guided by the teacher, submit & demonstrate the working result to the teacher for evaluating. This phase includes 3 iterations, each iteration would last for two weeks.
* ***Closing***: complete the software package and documentation, the phase lasts for two weeks, and would be completed with student teams’ presentation as final evaluation.

Project deliverables: the deliverable items for each phase of the project and relevant evaluation criteria are as in the table below. The details are defined in the topic syllabus

|  |  |  |
| --- | --- | --- |
| ***#*** | ***Phase*** | ***Deliverable Items & Evaluation Criteria*** |
| 1 | Initiation  *(Iteration 0)* | Product Backlog: this is just for team to plan the task assignments  SRS Document & Web Skeleton (5%)   * Overall contents as listed in the part I of the SRS document * Screen layouts of the system functions (listed in the part II) * Web Skeleton (HTML, CSS, JS files)   Design Document (Part I) (weighted 5%)   * Overall contents as listed in the part I of the Design Document * Vertical prototype: workable codes, acted as the code skeleton for the team to follow during the coding phase |
| 2 | Construction  *(Iteration 1)* | SRS Document (Part II): detailed contents for the relevant assigned functions of the iteration (weighted 1%)  Design Document (Part II & III): detailed contents for the relevant assigned functions of the iteration (weighted 1%)  Source Codes (weighted 8%):   * Code completeness: student would get maximum 60, 120 or 240 LOC for each completed function with the complexity of simple, medium or complex (respectively) * Code quality: graded by the teacher; the source code grade would be rated 100%, 75% or 50% of maximum LOC above, for the code quality of high, medium or low (respectively)   For each iteration, beside 10% evaluation of assigned works mastery, student would have 5% evaluation for professional working skills & attitude (details in the topic syllabus) |
| 3 | Construction  *(Iteration 2)* |
| 4 | Construction  *(Iteration 3)* |
| 5 | Closing  *(Final)* | Final project documentation & software package.  The evaluation criteria are as in the Construction phase, but for the whole project and with different weighting rates (see details in the topic syllabus) |

# II. Implementation Guides

## 1. Initiation Phase

In the first training session, student to register with the teacher about his/her programming language choice (Java or .NET, default Java)

Base on the student technical choice, teacher would arrange the working teams. Each team include 5-6 students and would work together to develop a software Web application as assigned / verified by the teacher.

Each team discusses & assign a team leader for them, then they together to allocate the assignments for each team member (by filling into the product backlog document) study the assigned requirements and then prepare solution for the team assignment.

Main tasks of the team during the Initiation phase:

* Select the team leader
* Arrange the function assignments & fill the product backlog
* Create Git environment on the Git environment:
  + Use GitLab (<https://gitlab.com>), Google sign-in using the @fpt.edu.vn account
  + Grant the relevant access right to the teacher & team members
  + Log the product backlog items into the GitLab
* Prepare the SRS document (the part I + screen layouts in the part II)
* Develop the Web Skeleton, which includes the header, footer, sider, and a demo content page for the Web that you would develop.
* Prepare the design document (the part I, which include code package/namespace, class naming conventions in each code package, and database schema)
* Develop the vertical prototype (technical/code skeleton) following the design you have made.

## 2. Construction Phase

The team member to start coding tasks as allocated (and verified by the teacher) during the initiation phase.

* Before coding, the team member has to prepare the detailed design (database, class, sequence) for relevant software function.
* The code must follow the provided coding convention & the design/naming conventions that the team has made
* After the coding, each team member list out all the rules (business logics, input validation, etc.) that he/she has applied & would test into the relevant function section in the SRS document

## 3. Closing Phase

The team complete/upgrade the project documents & source codes for the whole project following the guides as mentioned in the Construction Phase part.

# III. Other Guides/Requirements

The team discuss & agree on the working assignment (requirement function based) in the initiation phase. The assignment can be adjusted at the beginning of each phase or iteration

Templates to follow:

* Product Backlog: SWP391-AppDevProject\_Backlog Template
* SRS Document: SWP391-AppDevProject\_SRS Template
* Design Document: SWP391-AppDevProject\_Design Template
* Weekly Report: SWP391-AppDevProject\_Weekly Report

During the working progress:

* The team logs all the related tasks/issues & submit/track working results on the Git.
* By the end of each week, the team leader to consolidate all the working progress / status of each team member and the whole team to the weekly report to the teacher

All the submits to the teacher for evaluating/demo need to baselined/tagged via GitLab