

# Experience of developing Dividni plug-in for individual assignment

990708658 sshi145

## 1. Dividni plug-in develop experience

### 1.1 Assignment Requirement

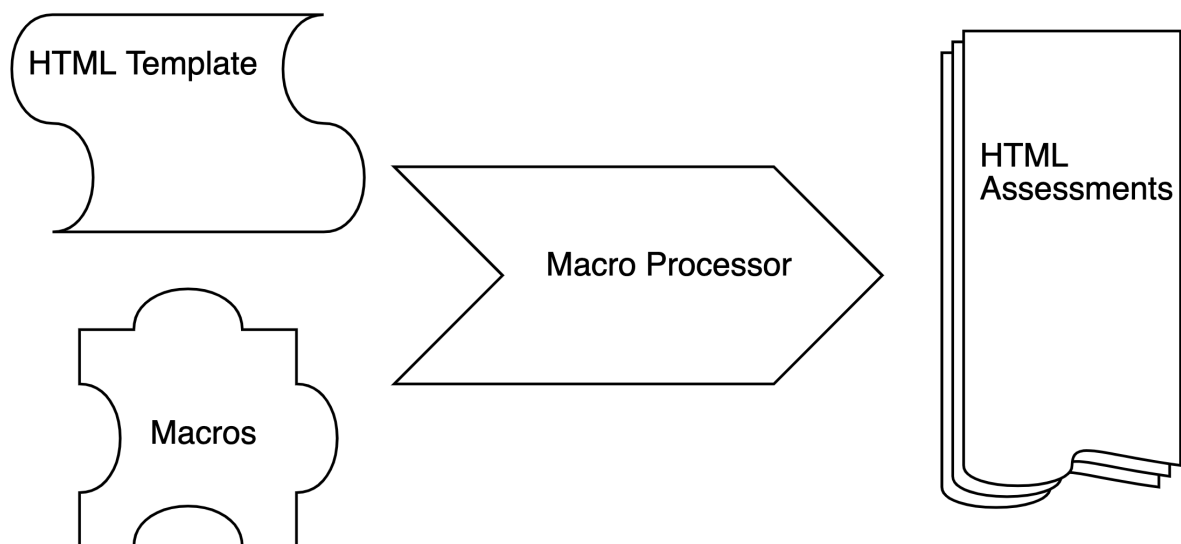
In this assignment, I need to find a way to create an individualized assignment for student use of a particular course[1]. For implementing, I need to develop a plug-in that would work with Dividni which is an individualized assessment framework.

### 1.2 Topic

For topic chosen, I chose CompSci 230 as the foundation course and basic Java coding practice as the specific topic. The sub-topics are about ArrayList operations, Regular expression, String splitting and Sorting. These operations are very commonly used in Java coding so that it helps students familiar with some basic understanding of Java.

### 1.3 Implementation

For implementation, plug-ins based on Dividni have architecture show on the graphic below[2]:



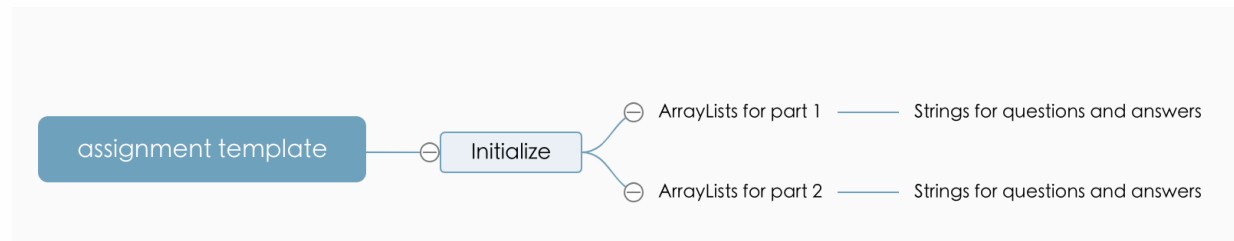
I need to implement an HTML assessment script and a macro separately. The macro contains methods which generate the questions and answers match to questions in the HTML template. Moreover, it also needs an auto-marking facility.

#### 1.3.1 Macro implementation

For developing the macro I need, steps are as follows:

1. Creating initialize-methods based on each student ID number. They give specific output from different IDs.
2. Creating methods used for the questions of assignments based on initialize-methods.
3. Creating methods used for showing correct answers and auto-marking.
4. Creating methods for auto-marking (auto extract answers from answer sheets students submitted, compare them with correct answers and give marks if they are correct).

These are easy to complete by using Dividni framework. I only need to generate strings required on the assignment questions, and the Dividni framework will put them on the location you assign each of them on the HTML template. The structure of method categories shows on the graph below:



There are also Auto-marking methods in Dividni. I need to override them. Firstly, I override the SubmittedAnswer method which extracts the answers student submitted. I added a regular expression to the method to remove all space in the submitted files. Then, I add mark rules to override the MarkingResult method. So that each submitted answer sheet can be auto-mark.

I writed two initial methods gener

### 1.3.2 Details in individual assignments

For individualizing assignments, I generate random numbers by each student ID in initialize-methods. Therefore, each student gets different numbers from each others', so that their answers will be different, which reduced plagiarism.

I wrote two initial methods generate different numbers based on the student ID. Then students need to write Java code for operations on these numbers in part 1. In part 2, I also use numbers from part 1 to generate some strings. Students need to extract useful information from these strings. These operations are basic and commonly used in coding. I believe students will have a useful assignment.

Each method named by the question id number. For example, the method return string of answer on question 1 named Q1. This makes the code readable.

### 1.3.3 HTML template implementation

The template of HTML in Dividni framework is similar to normal HTML file. There is a class called `cws_code_q` which need to put the methods generate strings for questions here. Similarly, there is a class called `cws_code_a` which is the locations of answers appears. When each assignment sheet is generated, the answers are hidden. After the marking process, the administrators can use the framework to send emails to students with correct answers.

I also use some block for emphasizing given inputs, tips and requirements. They have bright color so that students wont ignore them.

## 2. Literature review

Dorian explains the pure plug-in architectures in the article "*On Plug-ins and Extensible Architectures*"[3]. Based on this kind of structure, applications are made entirely of plug-ins. Dividni is this kind of things, it is a framework which works with macros and templates created by users. He presents that the applications based on the pure plug-in architectures respond quickly to rapid changes in requirements and upgradeability. When I use Dividni, I felt it have these abilities, the creation process for an assignment template is very fast and changeable. Each method in an implementation is very easy to change and update.

Dorian explains that an issue of plug-in-based applications is the installers and plug-in management because of the flexibility of this kind of applications. However, Dividni doesn't have this problem, it is as easy to install as other applications. More, if we import it to the code environment then we can use the framework.

Security in a plug-in environment is also an issue in the article. Dividni is a framework and developer use it to generate assignment templates which are pdf files and text files. Therefore, there are fewer security problems. Moreover, managing concurrent plug-in versions and dependencies is one of the main problems. Dividni does have many version, but we can use only one of them as a part of the developing environment.

However, Dorian point that convergence, compatibility, or interoperability of plug-in frameworks are considered. Dividni seems not good in these abilities. There is a little documentation about how to use it and how it can interact with other frameworks.

## Reference

- [1] A#2: Web Development. from: <https://canvas.auckland.ac.nz/courses/38256/assignments/134131>
- [2] Dividni. from: <https://dividni.com/>
- [3] Dorian Birsan. 2005. On Plug-ins and Extensible Architectures. *Queue* 3, 2 (March 2005), 40-46. DOI: <https://doi.org/10.1145/1053331.1053345>