AYU
Badkan - moodle
Software Design Document

Names :
Avi Boter,
Yair anisfeld
Uriel Sachs

Date: (13/01/2021)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

Moodle system is an online academic platform for both students and academic stuff for

studying , learning and communicating.

Badkan auto code tester is a free platform for academic stuff to apply class assignments to.

It receive a Github repository link and returns grade and feedback to the user.

Also, its can save the grade history of the student, which helps to the academic stuff to see his student progress.

Our main goal is to enable Moodle academic users to submit an academic code assignments with an automatic interface testers designed by the academic stuff

and create a plug-in that will enable to connect between the "Moodle" system and "Badkan" website which is an automatic tester.

1.2 Scope

- the plug-in will be design to work with an automatic interface tester "badkan" website
- the plug-in may support connection with other systems such the "Badkan" in this project scope.
- Our plugin will connect the student to the "Badkan" code runner.

As a result, in case of mistakes and issues with the student code, the student will get a chance to current its mistakes and learn better.

There for, an academic institutions will have the option to change there

teaching strategy.

1.3 Overview

This document will overview the design and architecture of the project.

1.4 Reference Material

Github
**Vision statement*
**Windows design document*
**SRS*

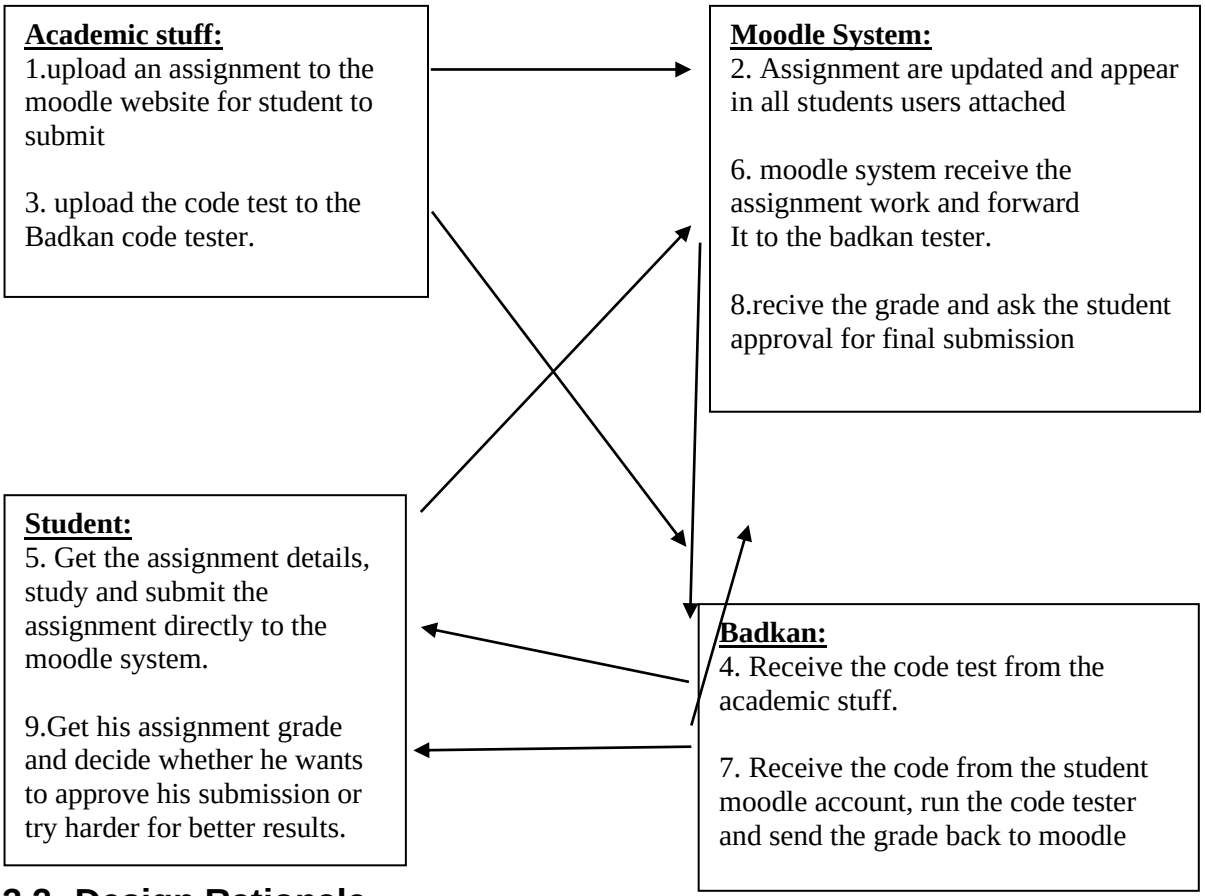
2. SYSTEM OVERVIEW

Our software is a plug-in feature installed on the Moodle software system.
As usually, academic stuff upload an assignment to the moodle website for student to submit, then, the student submit the assignment directly to the moodle system and automatically get the his submission grade from the badkan code tester and decide whether he wants to approve his submission or try harder for better results.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design and Decomposition Description

The next modular structure represents the relationship between the modules to achieve the complete functionality of the system.
the diagram showing the major subsystems and data repositories and their interconnections.



3.2 Design Rationale

We choose this architecture in that order (3.1) because of its logical order. Our implementation must be in these specific order due to the Academic Stuff/Lecturers and Students relations.

But also due to our willing, to let the student the opportunity to improve itself and its coding skills. First we allows the student to do a self examination, fix his code and learn from his mistakes. then we allow him to submit his assignments.

4. Data Design

4.1 Data Description

The database will be organized in a way that
each course will be a tree root that holds student relevant data as following:
Student id.
The Badkan details to login(e.g. password...)
URL(github)
temporary grade.

4.2 Data Dictionary

Course object -

Attributes:
Code Assignment file, Student array, course id , assignment id, lecturer id.

Student object -

Attributes:
id, GitHub URL , Badkan Login Details, Current grade , last code Review.

Functions:

Submit_to_badkan( ) -
will send the needs deatails to
badkan website and return the grade and feedback.

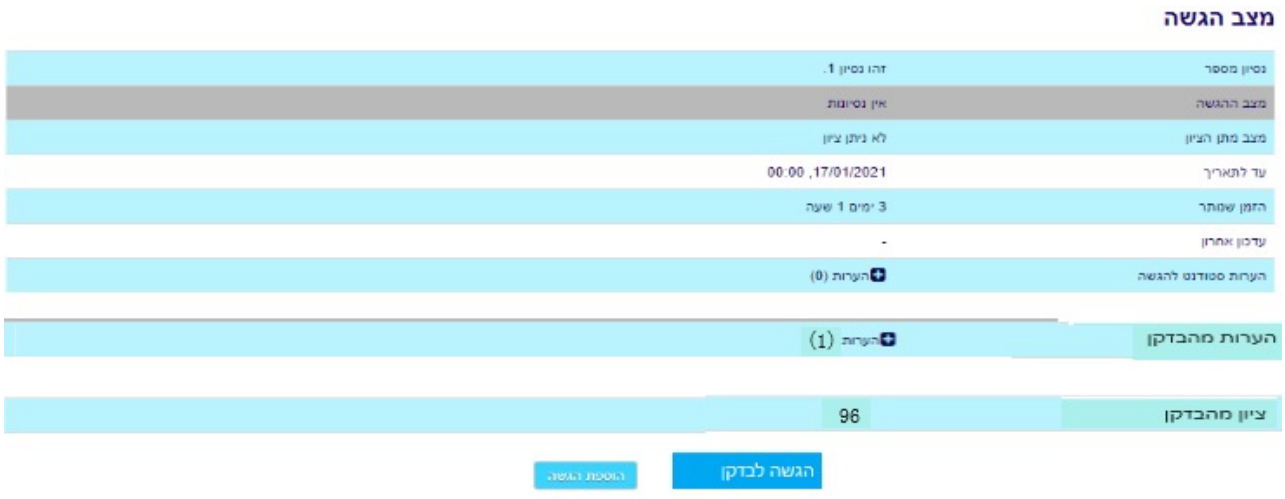
5. HUMAN INTERFACE DESIGN

5.1 Overview of User Interface

functionality of the system from the student perspective:
once the student enter the submission page of a task in the moodle the student will see an addition submission box called "badkan submission" which allow him to put the GitHub URL of his project and sent it to the badkan.
After the badkan process the results(grade and feedback/some error) the user will see a pop up window with the grade given and the feedback from the badkan(if an error occurred in the badkan running code there will be no grade but the user will still see the given output from the badkan). In that window the user will have the option to either save the grade as a "final grade" or take another try to improve his code and submit again.

5.2 Screen Images

some example of how the user interface with the plugin in the moodle
may look like
here we can see a case where the student sent his github URL in the "הגשה לבדקן"
box and got 96 grade and in the "הערות מהבדקן" he can see the feedback from the badkan.
If he would like to save and submit his grade he click the "הוספת הגשה".



6. REQUIREMENTS MATRIX

Component and data structure	Requirements from srs docoment
Student object – function 1 will need to get Student id, GitHub URL and his Badkan Login Details from. Course object – function 2 will get its values from in order to check validity. Same for function 3.	1.The Submission Box will enable every user to enter a "GitHub" repository link of his assignment code and deliver it to the Badkan grading system.
	2.In case of Broken link\not allowed link, user will get an error message elaborating and explaining the error source\reason and an help with solving it.
	3. The academic stuff will have to assign a unique Badkan-tester (Url) to the moodle assignment.
Submit_to_badkan( ) - function will receive data from the badkan and will return feedback to the Moodle system, one to be available both for student and Academic stuff. Also the function will receive Badkan review to the Moodle system, and then handle and seed it to the student.	the grade and feedback will be available via the moodle (it will be saved in a table in moodle).
	. There will be a separated window opened up which will contain the grade from the Badkan code tester.
	The academic stuff will have the ability to edit this (1) window
	The window will contain the full feedback from the Badkan code tester.