

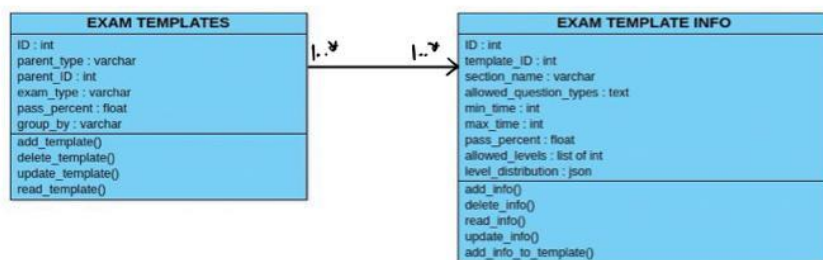
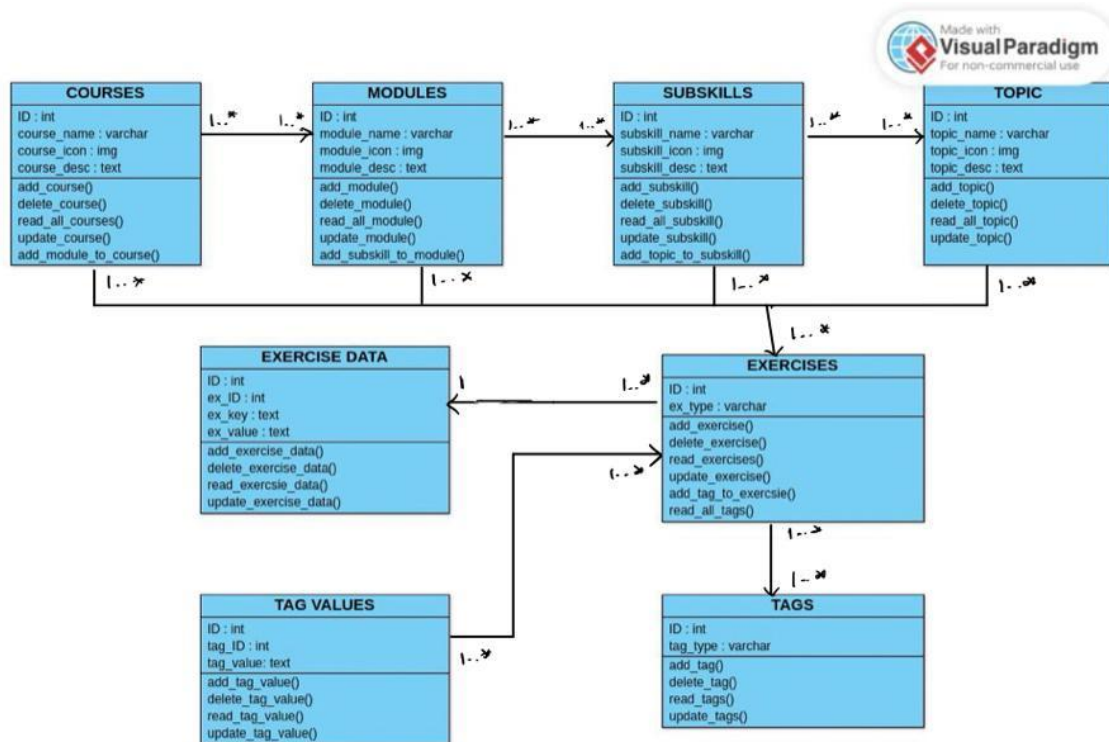
Product Design

Team 6, Recommendation Engine, Madireddy Ananya, Tejas Cavale, Aditya Mishra, Daksh Shah

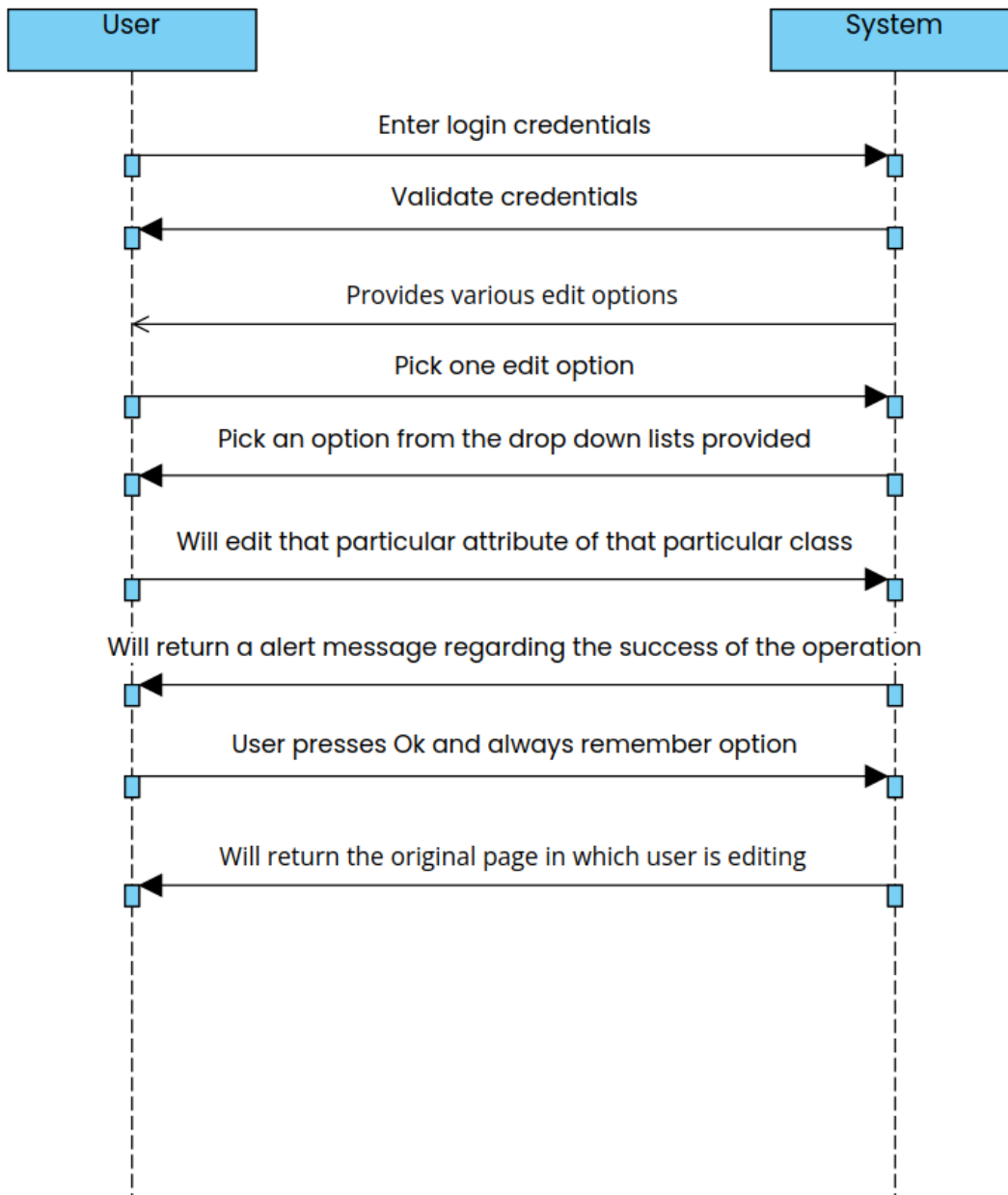
Design Model

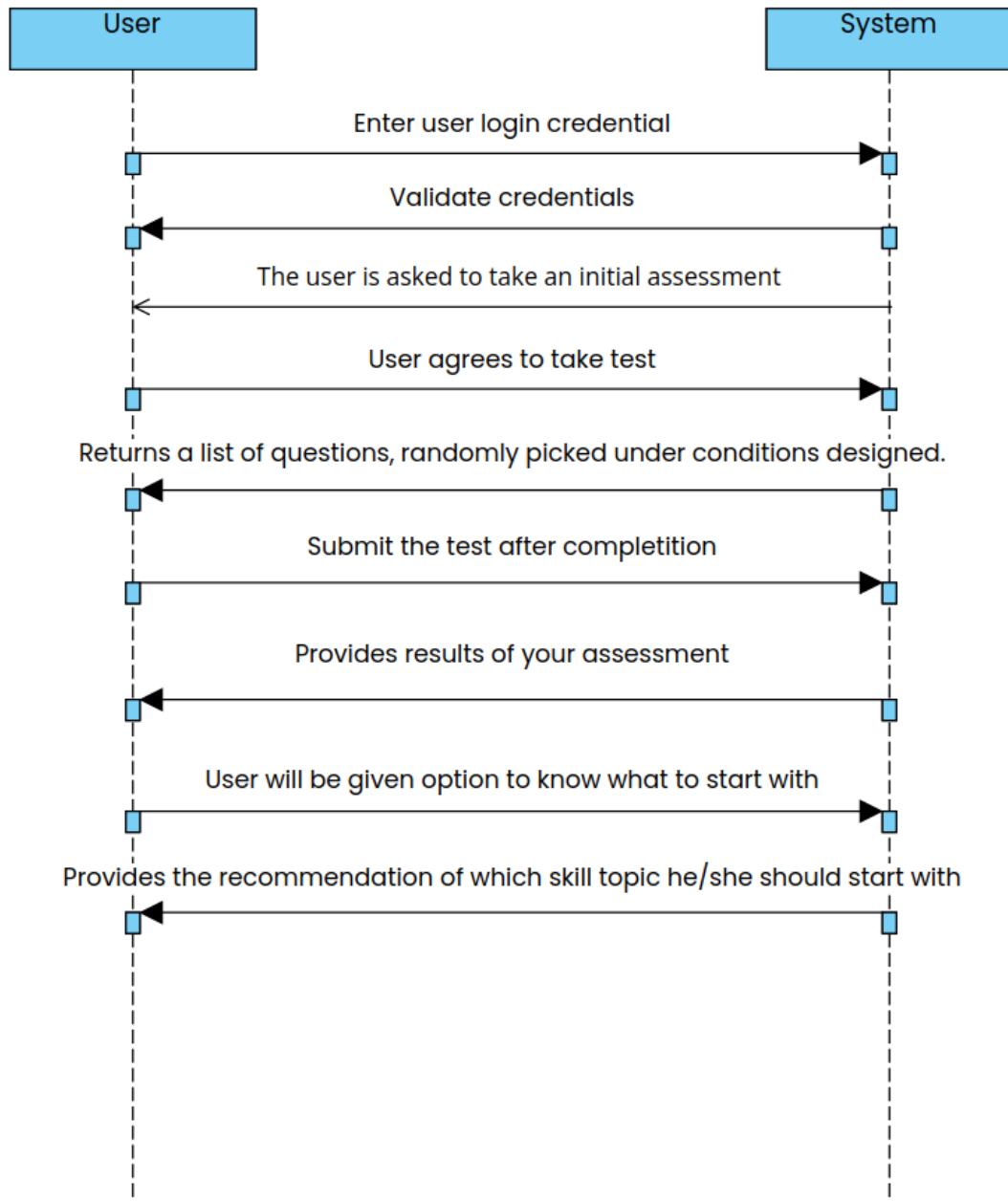
<Class No. 1> Courses	Class state <ul style="list-style-type: none">• Course id• Course name• Course icon• Course description Class behavior <ul style="list-style-type: none">• Add a course• Delete a course• Display all courses• Update a course attribute
<Class No. 2> Modules	Class state <ul style="list-style-type: none">• Module id• Module name• Module icon• Module description Class behavior <ul style="list-style-type: none">• Add a module• Delete a module• Display all modules• Update a module attribute• Add a module to a course
<Class No. 3> Sub skills	Class state <ul style="list-style-type: none">• Sub skill id• Sub skill name• Sub skill icon• Sub skill description Class behavior <ul style="list-style-type: none">• Add a sub skill• Delete a sub skill• Display all sub skills• Update a sub skill attribute• Add a sub skill to the module
<Class No. 4> Skill topics	Class state <ul style="list-style-type: none">• Skill topic id• Skill topic name• Skill topic icon• Skill topic description Class behavior <ul style="list-style-type: none">• Add a skill topic• Delete a skill topic

	<ul style="list-style-type: none"> • Display all skill topic • Update a skill topic attribute • Add a skill topic to sub skill
<Class No. 5> Exercises	Class state <ul style="list-style-type: none"> • Exercise id • Exercise type Class behavior <ul style="list-style-type: none"> • Add exercise • Delete exercise • Read exercise • Update exercise • Add tag to exercise • Display all the tags of that exercise
<Class No. 6> Exercise data	Class state <ul style="list-style-type: none"> • Id • Exercise id • Exercise key • Exercise value Class behavior <ul style="list-style-type: none"> • Add exercise data • Pick the key to add its value • Read exercise data • Update exercise data
<Class No. 7> Exam tempates	Class State <ul style="list-style-type: none"> • Id • Parent_type(Type of exam like unit test, course test, module test) • Parent id • Exam type • Pass_Percent • Group by Class behavior <ul style="list-style-type: none"> • Add a template • Delete template • Update attributes of all the templates
<Class No. 8> Exam template info	Class state <ul style="list-style-type: none"> • id • template id • section name • allowed question types • min_time for the exam • max time for the exam • pass percent • allowed levels • level distribution Class behavior <ul style="list-style-type: none"> • Add the attribute information • Adding the related questions • Update the pass percent • Compare with rules table



Sequence Diagram(s)





Design Rationale

- UI FOR CRUD OPERATIONS

Initially we designed the crud operations to function like as follows:

The user is going to give the course or module id and we are going to perform the crud operation and the search attribute for SQL query was id. Then we changed the searching parameter to the name of that class.

But we later rejected this idea, because Its hard to remember to the class id or name, so then we implemented a drop down list which shows all the class instances for a specific class, and the user picks one from the options, and then perform the necessary edit operation. It has much easier operation and has made crud operation feasible to use for the client as well.

- LINKING MODULE TO COURSES

Initially we had a thought of two different process.

To add a module to the list of modules and in the linkages table as well at the same time. Like adding a module, once the user picks a course and we are going to add under it.

The other approach was, to add the module normally to its table, and adding a module to a course should be carried out separately. Even after a long time.

M
o
d
u
l
e

a
d
d
i
t
i
o
n

h
a
s

n
o
t
h
i
n
g

t
o