

BỘ GIÁO DỤC VÀ ĐÀO TẠO TRƯỜNG ĐẠI HỌC PHENIKAA



-----*-----

BÁO CÁO ĐỒ ÁN CƠ SỞ

Đề tài: App Thi Trắc Nghiệm

Giảng viên: Vũ Đức Minh

Sinh viên: 19010061 - Dương Thị Thu Huyền

Mục Lục

Ý tưởng	4
Mô tả ý tưởng	4
b. Mục tiêu:	4
2.MVP của sản phẩm	5
a. Nghiên cứu thị trường	5
b.Giá trị sản phẩm	5
c. Các chức năng	5
d. Xây dựng	5
3. Thiết Kế	7
3.1 Database	7
3.2 Kiến trúc hệ thống	9
3.2.1 Kiến trúc	9
3.2.2 Use case	10
3.2.3 Đặc tả use case:	11
a.Đăng nhập	11
b.Đăng ký	12
c. Đăng xuất	13
d. Tạo bài test	14
e. Thêm câu hỏi vào bài test	15
f. Xóa câu hỏi trong bài test	16
g. Thêm câu hỏi vào question store	16
h. Xóa câu hỏi khỏi question store	17
k. Làm bài test	18
l. Xem lịch sử và xóa nộp bài	19
3.2.4 Sequence diagrams	20
4. Giao diện hệ thống	26
5. Đánh giá	33
6. Chức năng người sử dụng	33
Đóng góp	33

Giới thiệu

Những năm gần đây, giáo dục đã có rất nhiều sự thay đổi, đặc biệt là với sự xuất hiện của các phần mềm, ứng dụng. Bên cạnh việc học, giảng dạy trực tuyến, sử dụng các phần mềm thi online cũng ngày càng trở nên phổ biến hơn, nhất là trong thời điểm khó khăn do dịch bệnh như hiện nay. Để có thể tạo điều kiện cho sinh viên tiếp cận với nền tảng thi cử hiện đại và tiện lợi, cũng như các giáo viên có thể tiết kiệm hiệu quả thời gian và công sức thì có lẽ phần mềm thi trắc nghiệm trực tuyến chính là một công cụ hợp lý và hoàn hảo nhất.

Trong các kỳ thi, có hai hình thức thi phổ biến mà chúng ta vẫn thường thấy, đó chính là thi trắc nghiệm và thi tự luận. Đặc biệt hình thức thi trắc nghiệm ngày càng được ưa chuộng hơn do nó kiểm tra được kiến thức tổng quát thuộc nhiều khía cạnh, cũng như thời gian và cách chấm thi dễ dàng hơn so với hình thức tự luận. Đặc biệt càng có nhiều tổ chức, đơn vị áp dụng công nghệ vào hình thức thi trắc nghiệm này dù chỉ là bài kiểm tra nhỏ hay trong kỳ thi mang tính quốc tế như IELTS, JLPT...

1. Ý tưởng

a. Mô tả ý tưởng

Với tình hình dịch ngày càng rắc rối và khó lường trước như hiện nay thì việc học online sẽ gắn bó với học sinh trong thời gian khá dài nữa. Học online có ưu điểm là có thể học được ở mọi nơi, mà không cần gặp mặt, cô giáo vẫn có thể truyền tải được tin tức. Nhưng bên cạnh đó, học online cũng khó khăn với việc tiếp thu bài học hơn đối với học sinh, do một vài yếu tố khách quan như chưa có đủ thiết bị học tập, kết nối mạng không ổn định, ... Vì vậy đòi hỏi học sinh cần phải luyện tập những kiến thức đã học một cách có hệ thống, ví dụ như thông qua những bài test theo những chủ đề môn học. Do đó một ứng dụng làm test offline sẽ đáp ứng được nhu cầu học tập của học sinh, cũng như hỗ trợ việc giảng dạy của giáo viên. Ở đó, học sinh có thể tự do tham gia những bài test theo những chủ đề đa dạng kéo dài qua nhiều môn học, đồng thời giáo viên, học sinh cũng có thể tạo được bài test của riêng mình để hỗ trợ việc dạy học cũng như trao đổi thảo luận.

b. Mục tiêu:

- Tạo ra một ứng dụng dễ sử dụng
- Mang lại hiệu quả cho quá trình học tập và rèn luyện của người dùng
- Chia sẻ những bài test của mọi người dùng với nhau để làm phong phú chủ đề cũng như số lượng bài test
- Xây dựng app trên môi trường Desktop
- Ngôn ngữ lập trình: Java
- Quản lý cơ sở dữ liệu: mysql

2.MVP của sản phẩm

a. Nghiên cứu thị trường

Do nhu cầu học tập cũng như trau dồi kiến thức của mọi người ngày càng cao nên thị trường đang rất cần những phần mềm có thể giải quyết một cách hiệu quả các nhu cầu trên. Mặc dù cũng đã có rất nhiều sản phẩm làm về lĩnh vực này nhưng giao diện chưa thực sự bắt mắt và gần gũi với người dùng nên nhóm chúng em muốn tạo ra 1 sản phẩm có thể giải quyết được tồn tại đó.

b.Giá trị sản phẩm

- Môi trường cho mọi người luyện thi trắc nghiệm theo đa dạng chủ đề, môn học...
- Rèn luyện các kiến thức.

c. Các chức năng

High Priority	Medium Priority	Low Priority
<ul style="list-style-type: none"> - Đăng ký, đăng nhập tài khoản - Tạo danh sách câu hỏi - Tạo danh sách bài test - Chấm bài 	<ul style="list-style-type: none"> - Lưu trữ thông tin, kết quả người dùng 	<ul style="list-style-type: none"> - Xem lịch sử submit

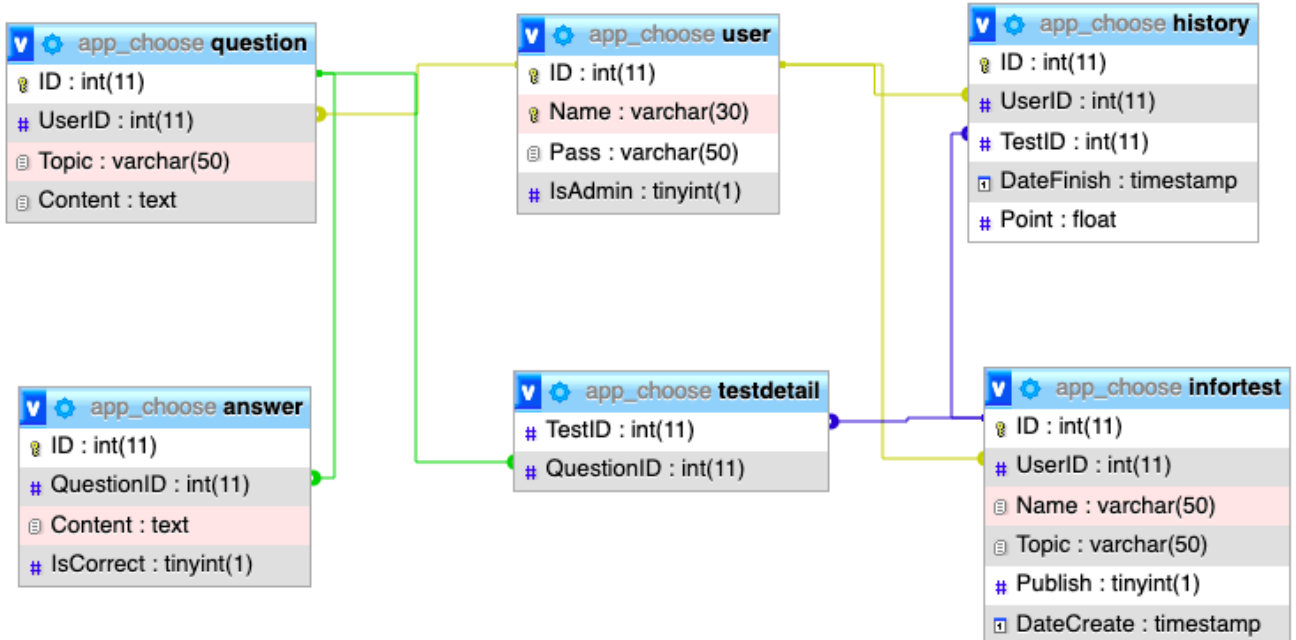
d. Xây dựng

- IDE: Apache NetBeans 12.5, IntelliJ IDEA Community Edition

- Hệ điều hành: Windows 10, Mac Osx
- Ngôn ngữ lập trình: java version 16
- Hệ quản trị cơ sở dữ liệu: Mysql

3. Thiết Kế

3.1 Database



Hình 3.1: Database

Mô tả database:

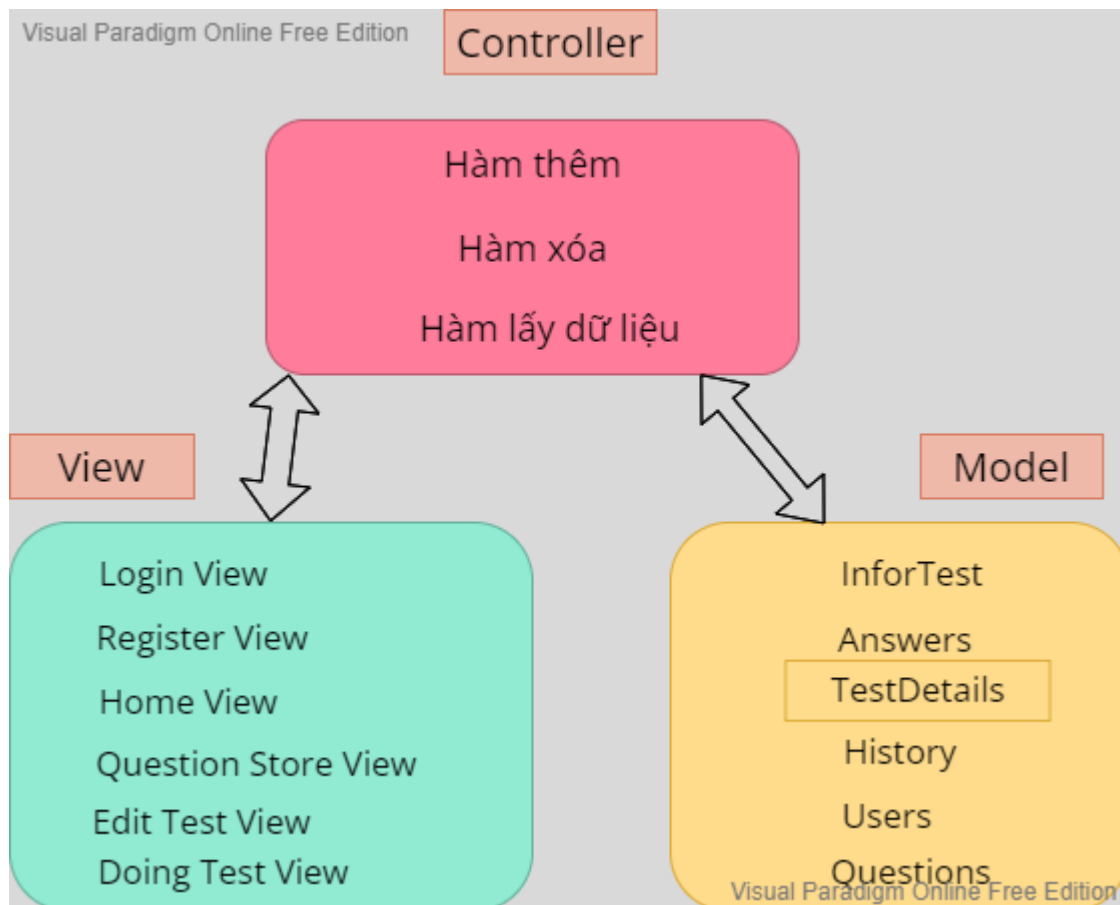
- Table User: lưu trữ thông tin tài khoản
 - o ID (người dùng): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - o Name: kiểu dữ liệu varchar(30), thuộc tính NOT NULL
 - o Pass: kiểu dữ liệu varchar(50), thuộc tính NOT NULL
 - o IsAdmin: kiểu dữ liệu boolean, thuộc tính Default false
- Table InforTest: lưu trữ thông tin chung của bài test
 - o ID (Test): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - o UserID (ID người dùng): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài User.ID
 - o Name: kiểu dữ liệu varchar(50), thuộc tính NOT NULL
 - o Topic: kiểu dữ liệu varchar(50), thuộc tính NOT NULL
 - o Publish: kiểu dữ liệu boolean, thuộc tính Default true

DateCreate: kiểu dữ liệu timestamp, thuộc tính NOT NULL,
Default current_timestamp()

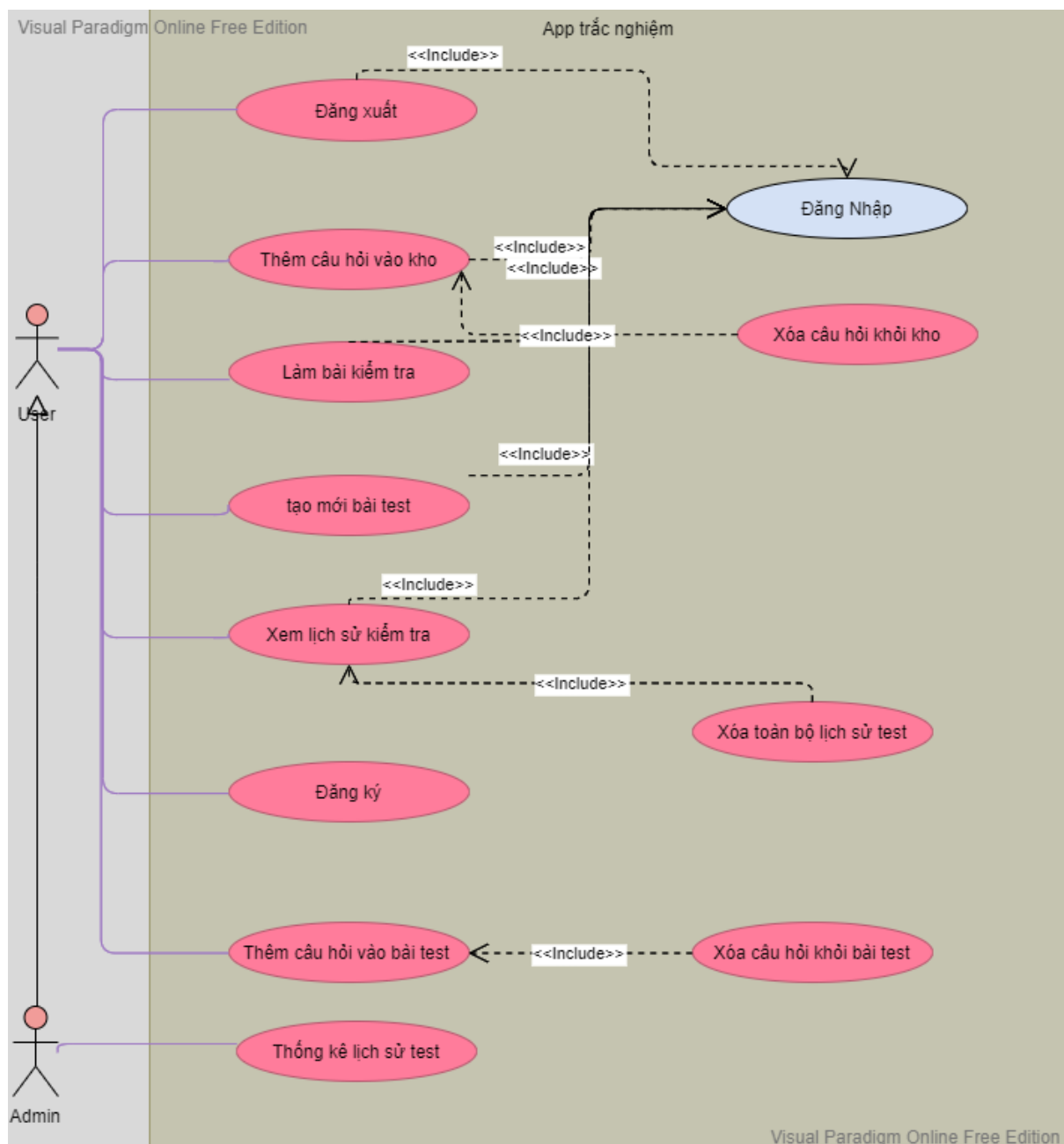
- Table Question: lưu trữ thông tin câu hỏi
 - o ID (Question): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - o UserID (ID người dùng): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài User.ID
 - o Topic: kiểu dữ liệu varchar(50), thuộc tính NOT NULL
 - o Content: kiểu dữ liệu text, thuộc tính NOT NULL
- Table TestDetail: lưu trữ thông tin câu hỏi của các bài test
 - o TestID (ID test): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài InforTest.ID
 - o QuestionID (ID câu hỏi): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài Question.ID
- Table Answer: lưu trữ thông tin câu trả lời tương ứng với ID question
 - o ID (Answer): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - o QuestionID (ID câu hỏi): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài Question.ID
 - o Content: kiểu dữ liệu text, thuộc tính NOT NULL
 - o IsCorrect: kiểu dữ liệu boolean, thuộc tính Default false
- Table History: lưu trữ lịch sử làm bài của User
 - o ID (history): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - o UserID (ID người dùng): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài User.ID
 - o TestID (ID test): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài InforTest.ID
 - o DateFinish: kiểu dữ liệu timestamp, thuộc tính NOT NULL, Default current_timestamp()
 - o Point: kiểu dữ liệu Float, thuộc tính NOT NULL, Default 0

3.2 Kiến trúc hệ thống

3.2.1 Kiến trúc

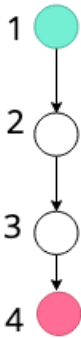


3.2.2 Use case

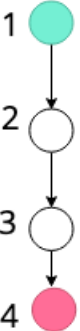


Hình 3.2: Use case diagram cho người dùng và người quản trị

3.2.3 Đặc tả use case:***a. Đăng nhập***

Use Case Name	- Đăng nhập
Description	- Là người dùng, tôi muốn đăng nhập vào ứng dụng để sử dụng
Actor(s)	- Người dùng, người quản trị hệ thống
Priority	- Must Have
Trigger	- Người dùng muốn đăng nhập vào sử dụng app
Pre-Condition(s):	<ul style="list-style-type: none"> - Tài khoản người dùng đã được tạo sẵn - Tài khoản người dùng đã được phân quyền - Thiết bị của người dùng đã được kết nối với database khi thực hiện đăng nhập
Post-Condition(s):	- Người dùng đăng nhập thành công
Basic Flow:  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) </pre>	<ol style="list-style-type: none"> 1. Người dùng truy cập ứng dụng. 2. Người dùng nhập tài khoản đã được tạo và chọn lệnh đăng nhập. 3. Hệ thống xác thực thông tin đăng nhập thành công. 4. Người dùng truy cập ứng dụng.
Exception Flow	- Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo, trả về giao diện trang chủ
Non-Functional Requirement	- Time-out cho màn hình đăng nhập dưới 20 giây

b.Đăng ký

Use Case Name	- Đăng ký
Description	- Là người dùng, tôi muốn đăng ký tài khoản để có thể sử dụng ứng dụng
Actor(s)	- Người dùng
Priority	- Must Have
Trigger	- Người dùng muốn đăng ký để sử dụng app
Pre-Condition(s):	<ul style="list-style-type: none"> - Tài khoản người dùng chưa được tạo sẵn - Thiết bị của người dùng đã được kết nối với database khi thực hiện đăng ký
Post-Condition(s):	- Người dùng đăng ký thành công, thông tin tài khoản được lưu vào database
<p>Basic Flow:</p>  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) </pre>	<ol style="list-style-type: none"> 1. Người dùng truy cập ứng dụng. 2. Người dùng chọn đăng ký để gửi yêu cầu đăng ký 3. Người dùng nhập thông tin tài khoản vào giao diện đăng ký, sau đó ấn đăng ký. 4. Hệ thống kiểm tra nếu tài khoản đã tồn tại, hoặc các trường nhập không khớp thì trả về thông báo yêu cầu nhập lại, còn không sẽ thêm tài khoản vào database đồng thời trả về

	thông báo đăng ký thành công và hiển thị giao diện đăng nhập
Exception Flow	<ul style="list-style-type: none"> - Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo. - Người dùng chưa nhập đầy đủ thông tin tài khoản khi đăng ký.
Non-Functional Requirement	<ul style="list-style-type: none"> - Time-out cho màn hình đăng ký dưới 20 giây

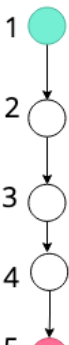
c. Đăng xuất

Use Case Name	- Đăng xuất
Description	- Là người dùng, tôi muốn đăng xuất khỏi tài khoản hiện hành.
Actor(s)	- Người dùng, người quản trị hệ thống
Priority	- Must Have
Trigger	- Người dùng muốn đăng xuất khỏi tài khoản đang sử dụng
Pre-Condition(s):	- Người dùng đã có tài khoản và thực hiện đăng nhập thành công
Post-Condition(s):	- Người dùng đăng xuất khỏi tài khoản và trở lại giao diện đăng nhập.
Basic Flow:	<p>1. Người dùng chọn đăng xuất và gửi yêu cầu đăng xuất.</p>



	<ol style="list-style-type: none"> 2. Hệ thống nhận được yêu cầu và thực hiện đăng xuất khỏi tài khoản hiện tại. 3. Quay về giao diện đăng nhập.
Non-Functional Requirement	<ul style="list-style-type: none"> - Time out cho màn hình Logout dưới 5 giây.

d. Tạo bài test

Use Case Name	- Tạo bài test mới
Description	- Là người dùng, tôi muốn dùng ứng dụng để tạo bài test mới
Actor(s)	- Người dùng
Priority	- Must Have
Trigger	- Người dùng muốn tạo bài test mới
Pre-Condition(s):	<ul style="list-style-type: none"> - Người dùng đã đăng nhập - Người dùng được phân quyền làm bài test - Thiết bị được kết nối với database
Post-Condition(s):	- Người dùng tạo thành công bài test mới
Basic Flow :	 <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) </pre> <ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng 2. Người dùng chọn tạo test mới 3. Hệ thống trả về form để tạo bài test 4. Người dùng điền xong thông tin, ấn submit

	5. Hệ thống lưu thông tin bài test vào database, và khi vào trang “Home” sẽ hiển thị bài test đó
Non-Functional Requirement	- Time out cho màn hình tạo test dưới 3 giây.

e. Thêm câu hỏi vào bài test

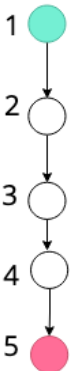
Use Case Name	- Chính sửa bài test
Description	- Là người dùng, tôi muốn dùng ứng dụng để chỉnh sửa bài test
Actor(s)	- Người dùng
Priority	- Must Have
Trigger	- Người dùng muốn chỉnh sửa bài test
Pre-Condition(s):	<ul style="list-style-type: none"> - Người dùng đã đăng nhập - Người dùng được phân quyền chỉnh sửa bài test - Thiết bị được kết nối với database
Post-Condition(s):	- Người dùng chỉnh sửa được bài test thành công
Basic Flow: <div data-bbox="300 1489 375 1870"> <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) </pre> </div>	<ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng 2. Người dùng chọn bài test muốn chỉnh sửa 3. Hệ thống trả về dữ liệu bài test 4. Người dùng click vào dòng câu hỏi trong bảng kho chứa câu hỏi, ấn nút “-> ” 5. Hệ thống Thêm câu hỏi đó vào bài test, đồng thời ở giao diện sẽ xóa đi câu hỏi đó trong bảng “Kho câu hỏi” và thêm câu hỏi đó vào bảng “Bài test”

f. Xóa câu hỏi trong bài test

Use Case Name	- Xóa bài test
Description	- Là người dùng, tôi muốn dùng ứng dụng để xóa bài test
Actor(s)	- Người dùng(khách hàng)
Priority	- Phải có
Trigger	- Người dùng muốn xóa bài test
Pre-Condition(s):	<ul style="list-style-type: none"> - Người dùng đã đăng nhập - Người dùng được phân quyền xóa bài test - Thiết bị được kết nối với database
Post-Condition(s):	- Người dùng xóa được bài test thành công
Basic Flow	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) </pre> </div> <div> <ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng 2. Người dùng chọn bài test muốn chỉnh sửa 3. Hệ thống trả về dữ liệu bài test 4. Người dùng click vào dòng câu hỏi trong bảng chứa câu hỏi trong bài test, ấn nút “<-” 5. Hệ thống xóa câu hỏi đó khỏi bài test, đồng thời ở giao diện sẽ xóa đi câu hỏi đó trong bảng “Bài test” và thêm câu hỏi đó vào bảng “Kho câu hỏi” </div> </div>

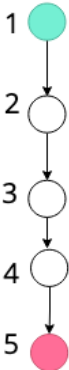
g. Thêm câu hỏi vào question store

Use Case Name	- Thêm câu hỏi mới vào question store
---------------	---------------------------------------

Description	- Là người dùng, tôi muốn dùng ứng dụng để thêm câu hỏi vào database của mình
Actor(s)	- Người dùng
Priority	- Must Have
Trigger	- Người dùng muốn thêm câu hỏi mới
Pre-Condition(s):	- Người dùng đã đăng nhập - Thiết bị được kết nối với database
Post-Condition(s):	- Người dùng thêm câu hỏi thành công
Basic Flow :	<ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng 2. Người dùng chọn question store 3. Hệ thống trả về form để tạo câu hỏi mới và hiện thị list các câu hỏi đã có sẵn trước đây 4. Người dùng điền xong thông tin, ấn Add 5. Hệ thống lưu thông tin câu hỏi vào database và ở giao diện, tự động câu hỏi đó được thêm vào bảng “Kho câu hỏi”  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) </pre>
Non-Functional Requirement	- Time out cho màn hình tạo câu hỏi dưới 3 giây.

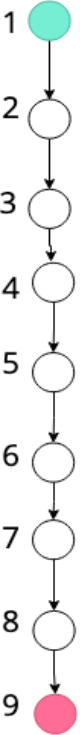
h. Xóa câu hỏi khỏi question store

Use Case Name	- Xóa câu hỏi khỏi question store
Description	- Là người dùng, tôi muốn xóa câu hỏi khỏi database của mình
Actor(s)	- Người dùng

Priority	- Must Have
Trigger	- Người dùng muốn xóa câu hỏi cũ
Pre-Condition(s):	<ul style="list-style-type: none"> - Người dùng đã đăng nhập - Thiết bị được kết nối với database
Post-Condition(s):	- Người dùng xóa câu hỏi thành công
<p>Basic Flow :</p>  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) </pre>	<ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng 2. Người dùng chọn question store 3. Hệ thống trả về form để tạo câu hỏi mới và hiện thị list các câu hỏi đã có sẵn trước đây 4. Người dùng chọn câu hỏi muốn xóa trong list câu hỏi, ấn Xóa 5. Hệ thống cập nhật database và tại giao diện tự động xóa câu hỏi đó khỏi bảng “Kho câu hỏi”
Non-Functional Requirement	- Time out cho màn hình xóa câu hỏi dưới 3 giây.

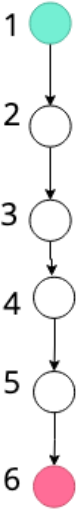
k. Làm bài test

Use Case Name	- Làm bài test
Description	- Là người dùng, tôi muốn tham gia làm các bài test đang hiển thị trên giao diện trang chủ
Actor(s)	- Người dùng, người quản trị hệ thống
Priority	- Must Have
Trigger	- Người dùng muốn tham gia làm các bài test
Pre-Condition(s):	- Người dùng đã có tài khoản

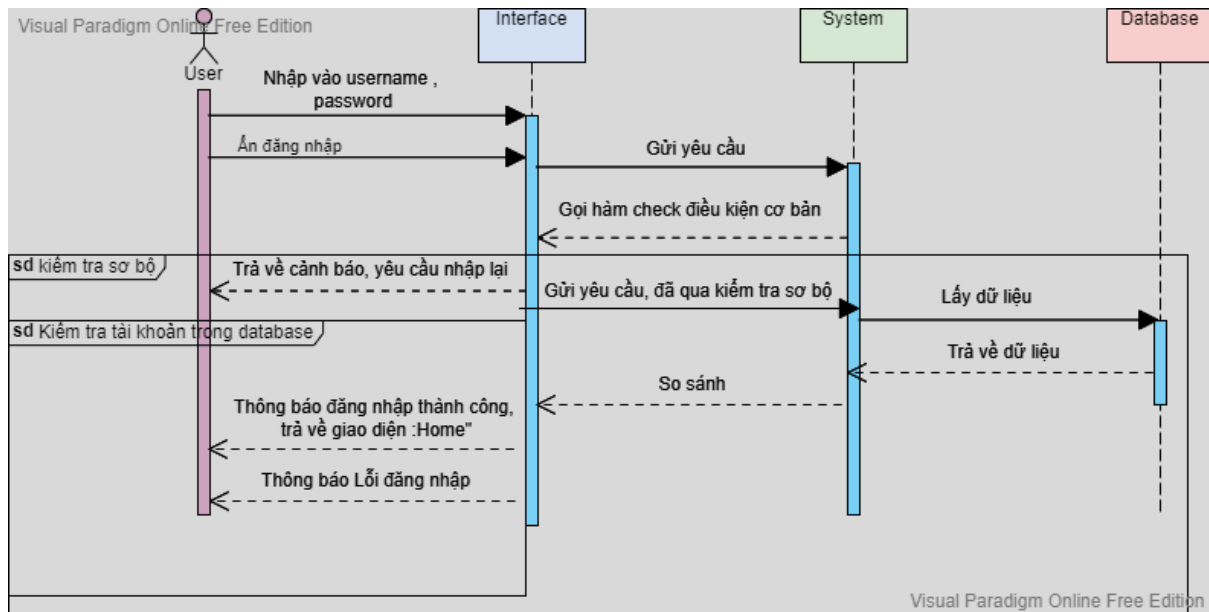
	<ul style="list-style-type: none"> - Người dùng đăng nhập thành công - Tài khoản có quyền tham gia làm bài test - Thiết bị đã kết nối tới database
Post-Condition(s):	<ul style="list-style-type: none"> - Người dùng tham gia làm bài test thành công - Sau khi hoàn thành hệ thống sẽ chấm điểm bài test và thông báo điểm rồi lưu vào lịch sử
<p>Basic Flow:</p>  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) 5 --> 6((6)) 6 --> 7((7)) 7 --> 8((8)) 8 --> 9((9)) </pre>	<ol style="list-style-type: none"> 1. Người dùng truy cập vào ứng dụng 2. Người dùng nhập thông tin tài khoản 3. Người dùng click vào button Login 4. Hệ thống xác nhận đăng nhập thành công và chuyển đến giao diện trang chủ 5. Người dùng click vào button play trong bài test muốn tham gia 6. Hệ thống load data và hiển thị giao diện làm bài 7. Người dùng trả lời hết các câu hỏi và click và submit 8. Nếu có câu chưa được chọn, hệ thống trả về thông báo yêu cầu người dùng phải chọn hết còn không hệ thống trả về giao diện kết quả 9. Nếu người dùng ấn “Save” , thì sẽ lưu lại lịch sử bài test này
Non-Functional Requirement	<ul style="list-style-type: none"> - Time out cho màn hình submit dưới 60 giây.

1. Xem lịch sử và xóa nộp bài

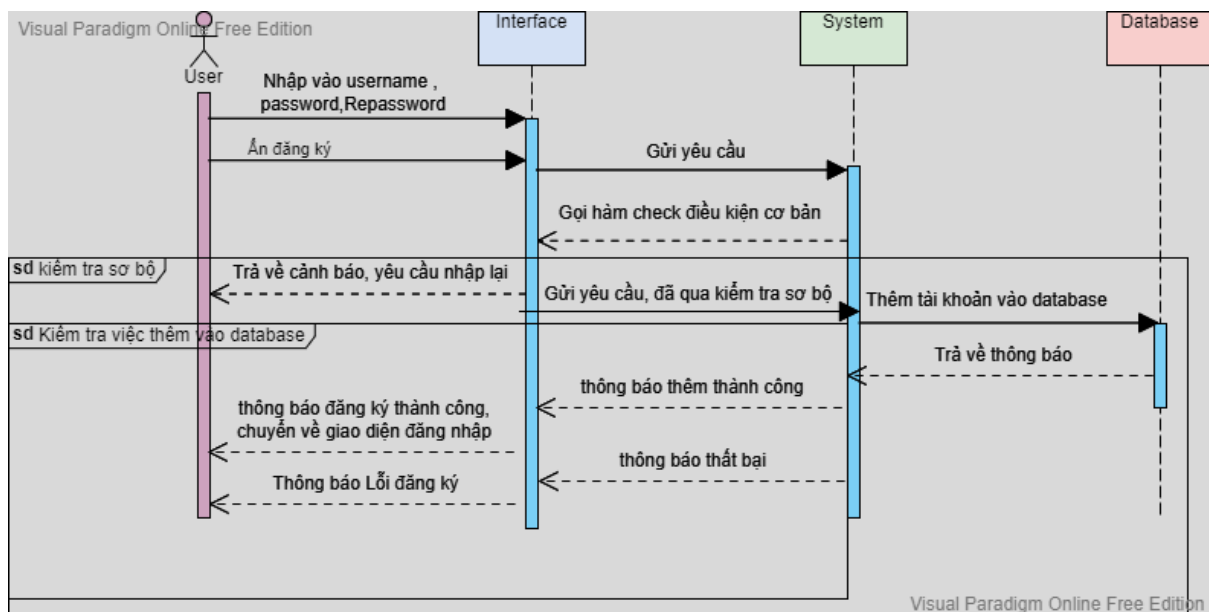
Use Case Name	- Xem lịch sử nộp bài
---------------	-----------------------

Description	- Là người dùng, tôi muốn xem lại lịch sử tôi đã tham gia các bài test
Actor(s)	- Người dùng, người quản trị hệ thống
Priority	- Must Have
Trigger	- Người dùng muốn xem lại lịch sử nộp bài
Pre-Condition(s):	<ul style="list-style-type: none"> - Người dùng đã có tài khoản - Người dùng đăng nhập thành công - Thiết bị đã kết nối tới database
Post-Condition(s):	- Hệ thống hiển thị lịch sử nộp bài của người dùng
<p>Basic Flow:</p>  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) 5 --> 6((6)) </pre>	<ol style="list-style-type: none"> 1. Người dùng truy cập vào ứng dụng 2. Người dùng nhập thông tin tài khoản 3. Người dùng click vào button Login 4. Hệ thống xác nhận đăng nhập thành công và chuyển đến giao diện trang chủ 5. Người dùng click vào button History, Hệ thống load data và hiển thị lịch sử làm bài của người dùng 6. Nếu người dùng nhấn “Delete all” hệ thống sẽ xóa toàn bộ lịch sử test của người dùng đó trong database cũng như trong bảng hiển thị ở giao diện

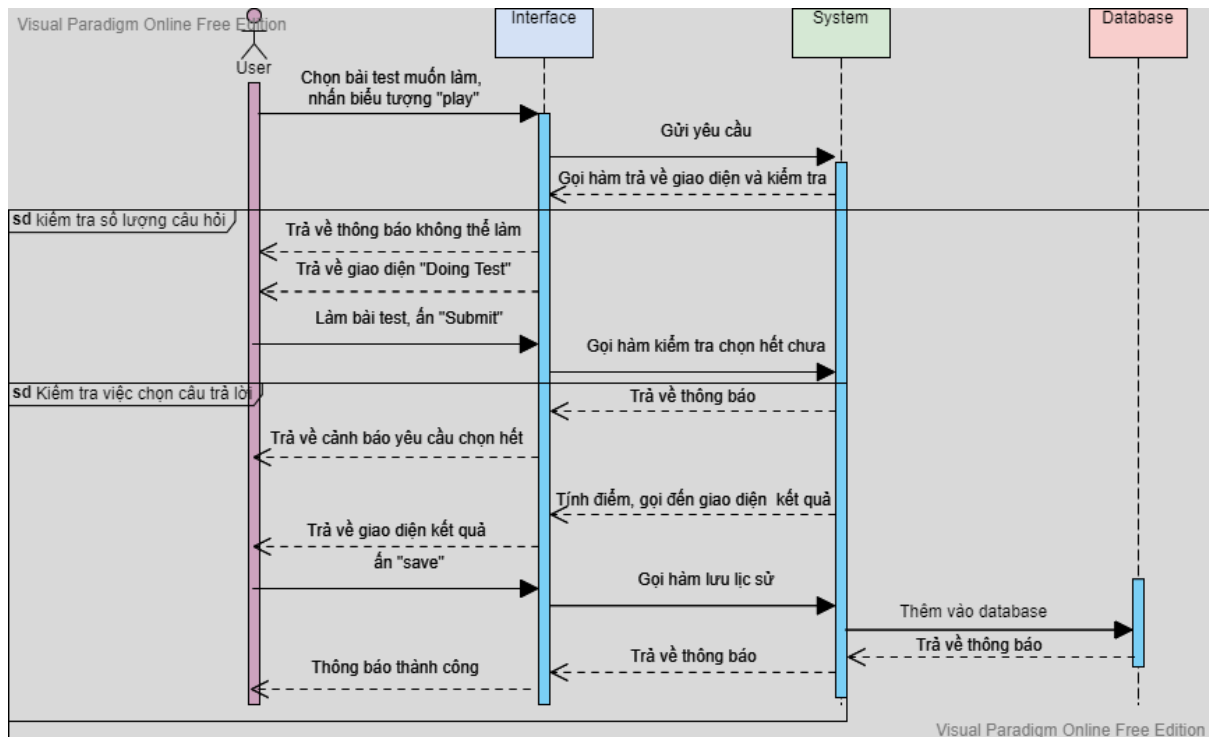
3.2.4 Sequence diagrams



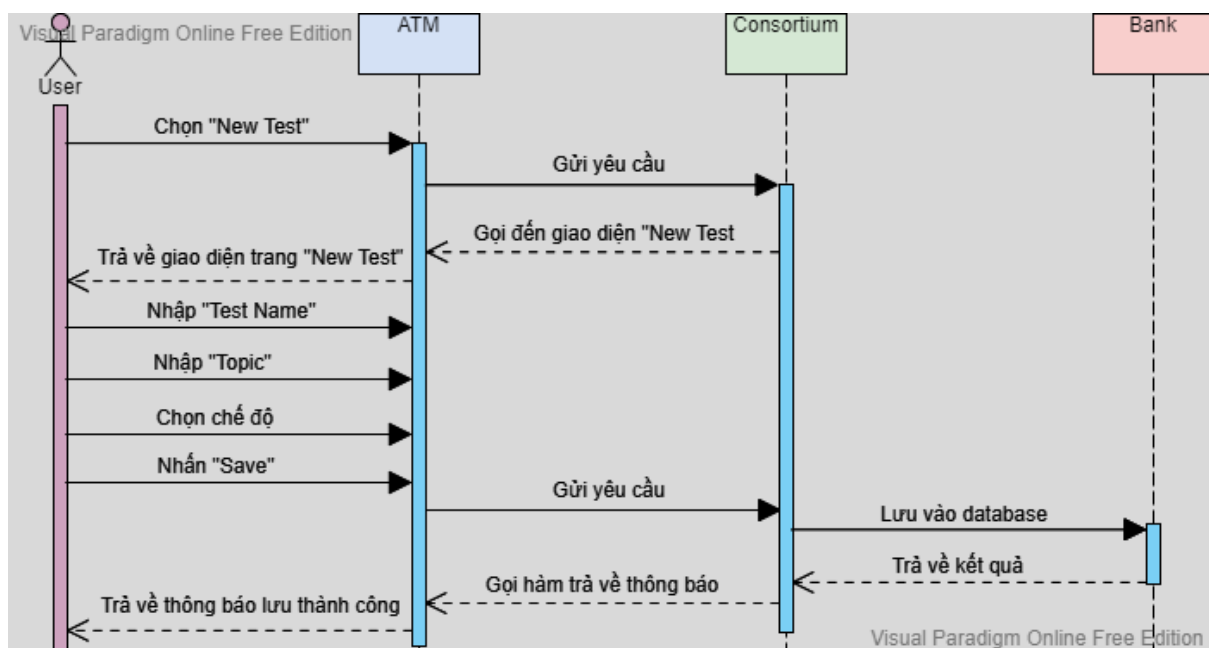
Hình 1: Sequence diagrams Login



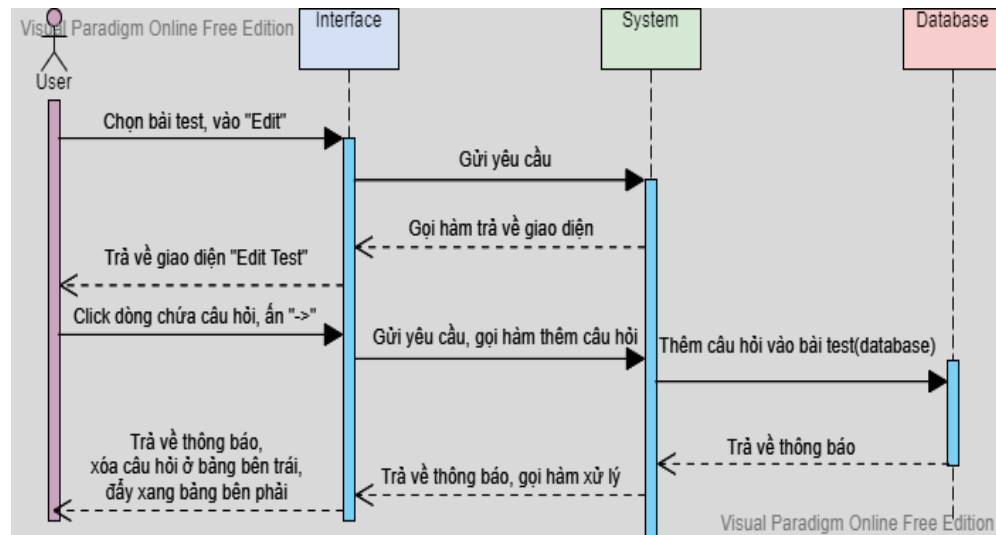
Hình 2: Sequence diagrams cho chức năng Đăng ký



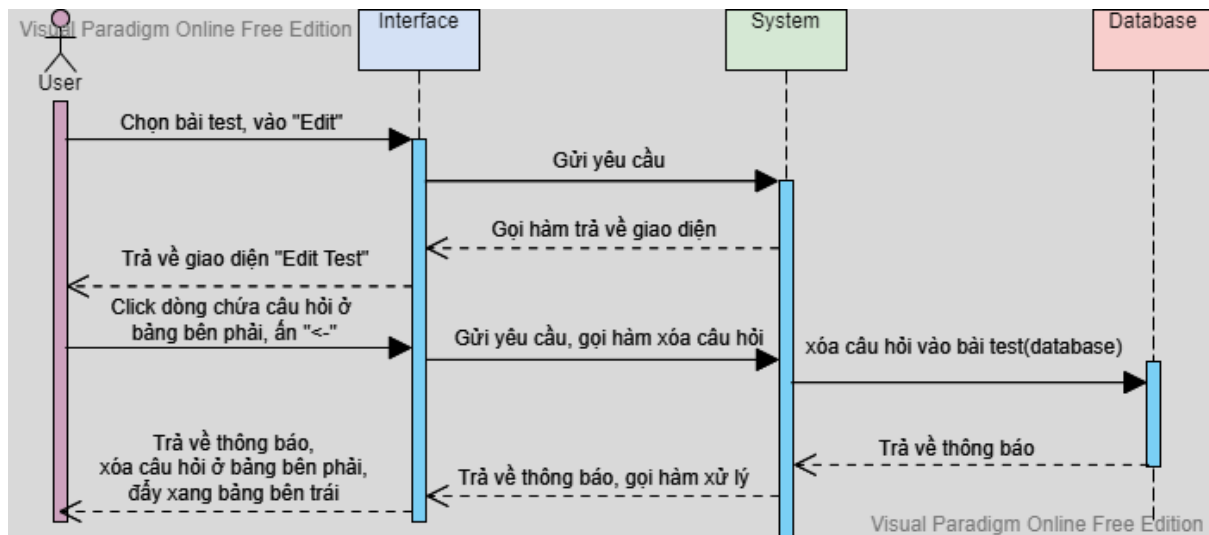
Hình 3: Sequence diagrams cho chức năng làm kiểm tra



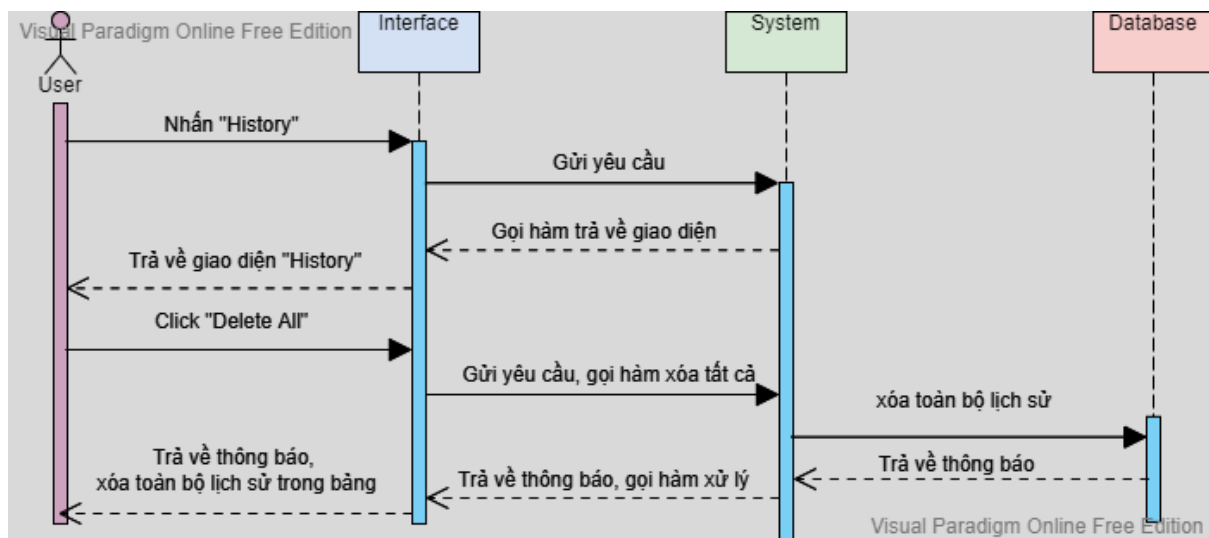
Hình 4: Sequence diagrams cho chức năng tạo bài kiểm tra mới



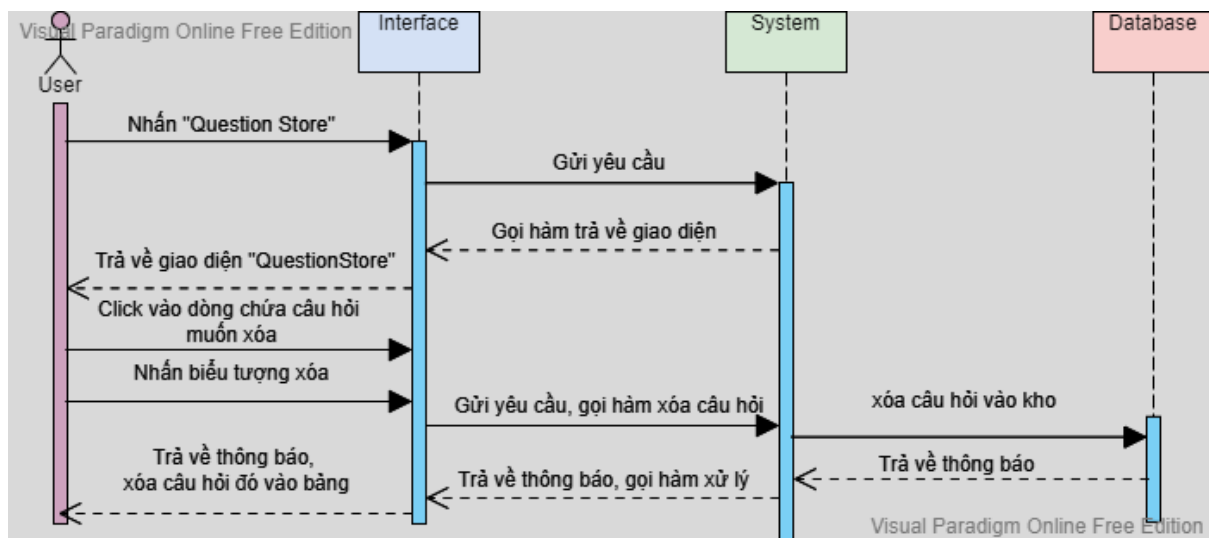
Hình 5: Sequence diagrams cho chức năng thêm câu hỏi vào bài test



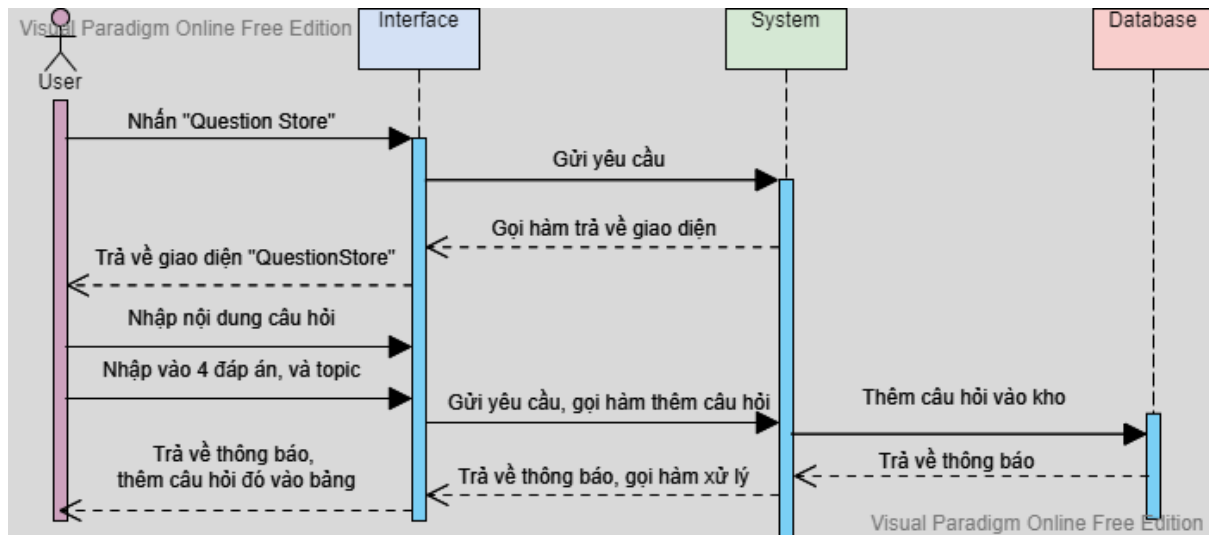
Hình 6: Sequence diagrams xóa câu hỏi khỏi bài test



Hình 7: Sequence diagrams cho chức năng xóa toàn bộ lịch sử làm test



Hình 8: Sequence diagrams xóa câu hỏi trong kho câu hỏi



Hình 9: Sequence diagrams thêm câu hỏi vào kho câu hỏi

4. Giao diện hệ thống

a. Đăng ký

Quizz App

Register

UserName

Tên đăng nhập

Password

RePassword

Cancel

Register

b. Đăng nhập

Quiz app

Hi huyen!

+ New Test

Question Store

Log Out

History

kt 1 tiết

2 Question

Toán

11/25/21, 2:01 AM

Publish

▶

Kt 30 phút

1 Question

Văn

11/25/21, 2:01 AM

Publish

▶

kt 1 tiet

1 Question

english

11/28/21, 1:41 AM

Publish

▶

kt

0 Question

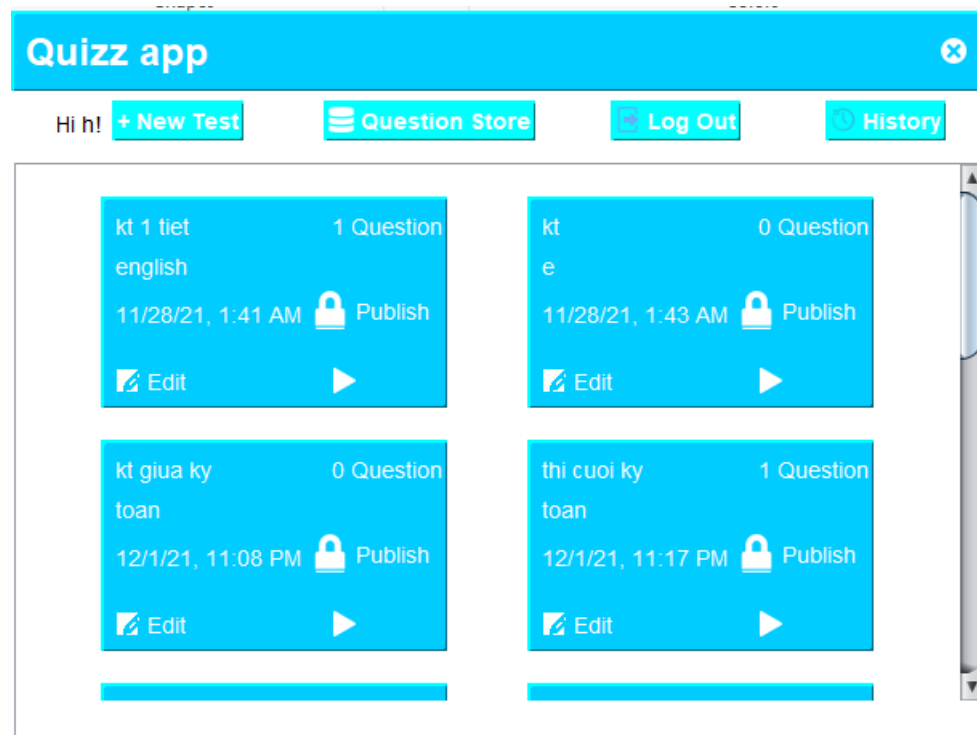
e

11/28/21, 1:43 AM

Publish

▶

d. Trang chủ dành cho người dùng



e. Tạo bài test

Quizz app

Create a new test

Test name

Topic

Publish

☐ Publish
 ☐ Private

Save

f. Thêm/xóa câu hỏi trong kho dữ liệu

≡ Question store

Home

Close

Create A New Question

Question:

Correct answer

False answer

False answer

False answer

Topic



Add Question

ID	Question	Correct Answer	Numbers Of Ans...	Topic
17	toan	1-10	-9	4

g. Lịch sử nộp bài

History  				
History 				
Test Name	Topic Name	Number Of Questio...	Test Date	Point

h. Chỉnh sửa bài test

Edit Test  													
kt 1 tiet	english 11/28/21, 1:41 AM												
Add Question													
<table border="1"> <thead> <tr> <th colspan="2">Question In Store</th> </tr> <tr> <th>ID</th> <th>QUESTION</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>4-5</td> </tr> </tbody> </table>	Question In Store		ID	QUESTION	5	4-5	<table border="1"> <thead> <tr> <th colspan="2">Question In Test</th> </tr> <tr> <th>ID</th> <th>CONTENT</th> </tr> </thead> <tbody> <tr> <td>17</td> <td>1-10</td> </tr> </tbody> </table>	Question In Test		ID	CONTENT	17	1-10
Question In Store													
ID	QUESTION												
5	4-5												
Question In Test													
ID	CONTENT												
17	1-10												
<div>➡</div> <div>⬅</div>													

i. Làm bài test

Doing Test

Test: kt 1 tiet
Topic: toan

Question 1: 1+5

☒ 8

☐ 6

☐ 7

☐ 3

ii. Thống kê lịch sử làm test đối với admin

Quizz app				
ID	UserID	TestID	DateFinish	Point
1	4	18	2021-12-08 02:1...	10.0

5. Đánh giá

- 5.1 Chương trình đã đáp ứng được những yêu cầu đặt ra
- 5.2 Đảm bảo tính dễ dùng
- 5.3 Đảm bảo tính nhất quán của dữ liệu
- 5.4 Hoạt động ổn định
- 5.5 Tốn ít tài nguyên

6. Chức năng người sử dụng

❖ Các chức năng xây dựng thành công:

- Đăng nhập/ đăng xuất
- Đăng ký
- Tạo kho chứa câu hỏi và đáp án
- Tạo và chỉnh sửa bài test
- Tham gia làm bài
- Hiển thị lịch sử làm bài

7. Đóng góp

Code backend và giao diện.

