



TRƯỜNG ĐẠI HỌC PHENIKAA
KHOA CÔNG NGHỆ THÔNG TIN



Giao Diện Người Máy
< App thi trắc nghiệm >

Giảng viên hướng dẫn : **Vũ Quang Dũng**

Sinh viên thực hiện : 19010042 – Nguyễn Hữu Sang
19010061 – Dương Thị Thu Huyền
19010017 – Phan Việt Linh
19010016 – Nguyễn Trung Kiên
19010029 – Nguyễn Thị Thanh



Hà Nội - 2021

Table of Contents

<i>I. Giới thiệu</i>	3
1. Mô tả ý tưởng.....	3
2. Mục tiêu:	4
<i>II. MVP của sản phẩm</i>	4
1. Nghiên cứu thị trường	4
2. Giá trị sản phẩm	4
3. Các chức năng.....	5
4. Xây dựng.....	5
<i>III. Thiết Kế</i>	6
1. Database	6
2. Kiến trúc hệ thống.....	8
2.1. Architecture.....	8
2.2. Use case.....	9
2.3. Sequence diagrams.....	18
3. Giao diện hệ thống	25
<i>IV. Đánh giá</i>	30
<i>V. Chức năng người sử dụng</i>	30
<i>VI. Đóng góp của từng thành viên</i>	30

Github: [App](#)

Video demo: [Demo](#)

I. 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, TOEFLS Ibt...

Ý tưởng:

1. 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.

2. 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

II. MVP của sản phẩm

1. 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.

2. 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.

3. Các chức năng

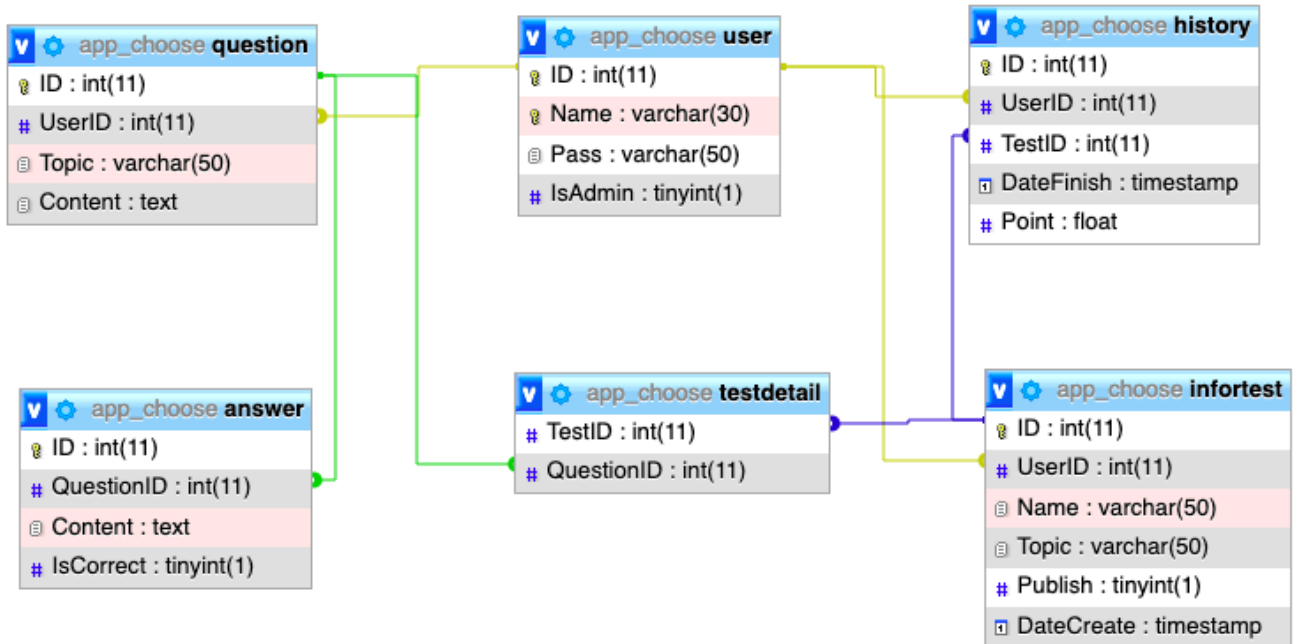
High Priority	Medium Priority	Low Priority
<ul style="list-style-type: none">- Đăng ký, đăng nhập tài khoản- Tạo danh sách câu hỏi- Tạo danh sách bài test- Chấm bài	<ul style="list-style-type: none">- Lưu trữ thông tin, kết quả người dùng	<ul style="list-style-type: none">- Xem lịch sử submit

4. 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

III. Thiết Kế

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

- *Table Question*: lưu trữ thông tin câu hỏi
 - ID (Question): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - UserID (ID người dùng): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài User.ID
 - Topic: kiểu dữ liệu varchar(50), thuộc tính NOT NULL
 - Content: kiểu dữ liệu text, thuộc tính NOT NULL

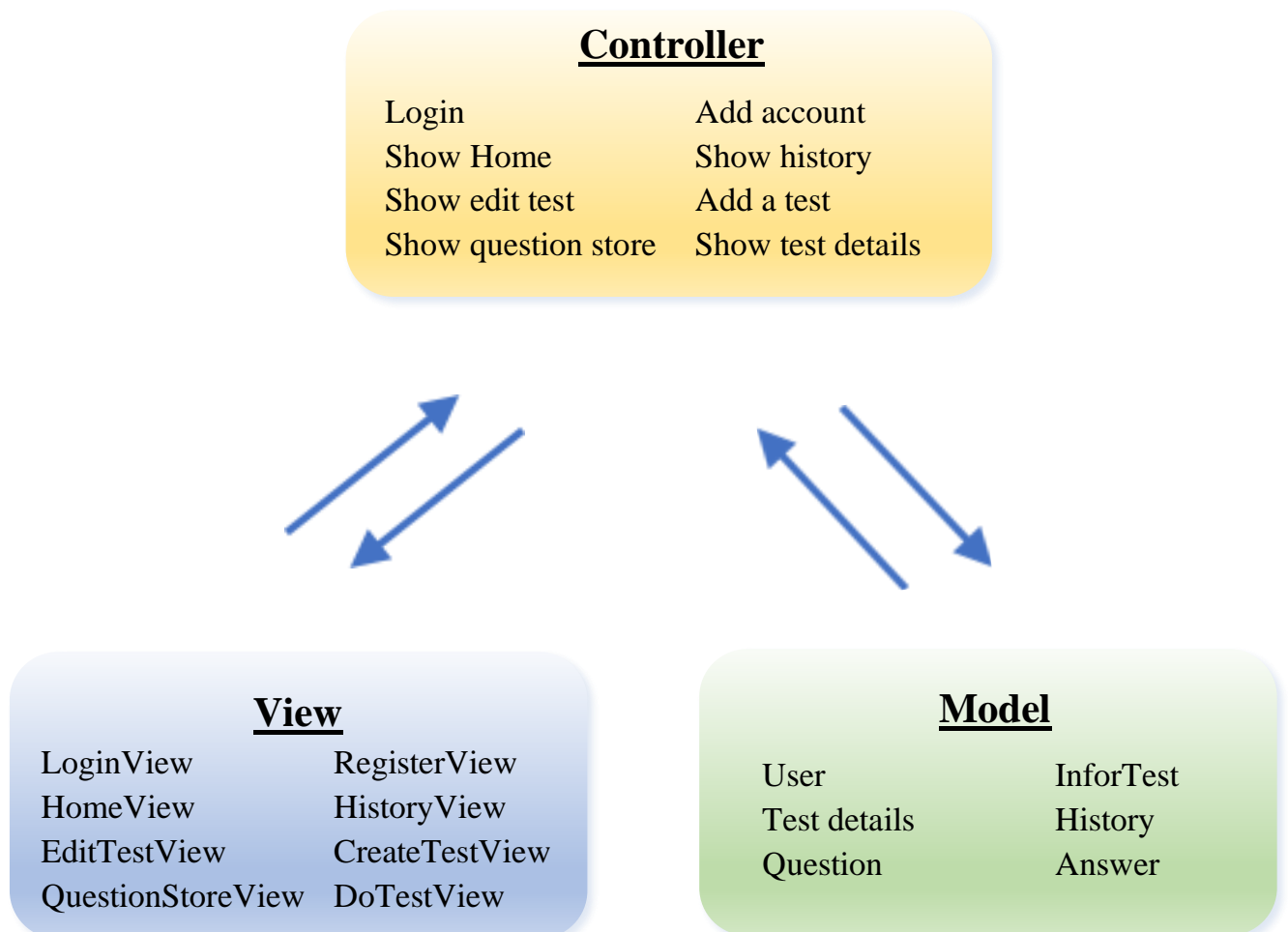
- *Table TestDetail*: lưu trữ thông tin câu hỏi của các bài test
 - TestID (ID test): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài InforTest.ID
 - QuestionID (ID câu hỏi): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài Question.ID

- *Table Answer*: lưu trữ thông tin câu trả lời tương ứng với ID question
 - ID (Answer): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - QuestionID (ID câu hỏi): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài Question.ID
 - Content: kiểu dữ liệu text, thuộc tính NOT NULL
 - IsCorrect: kiểu dữ liệu boolean, thuộc tính Default false

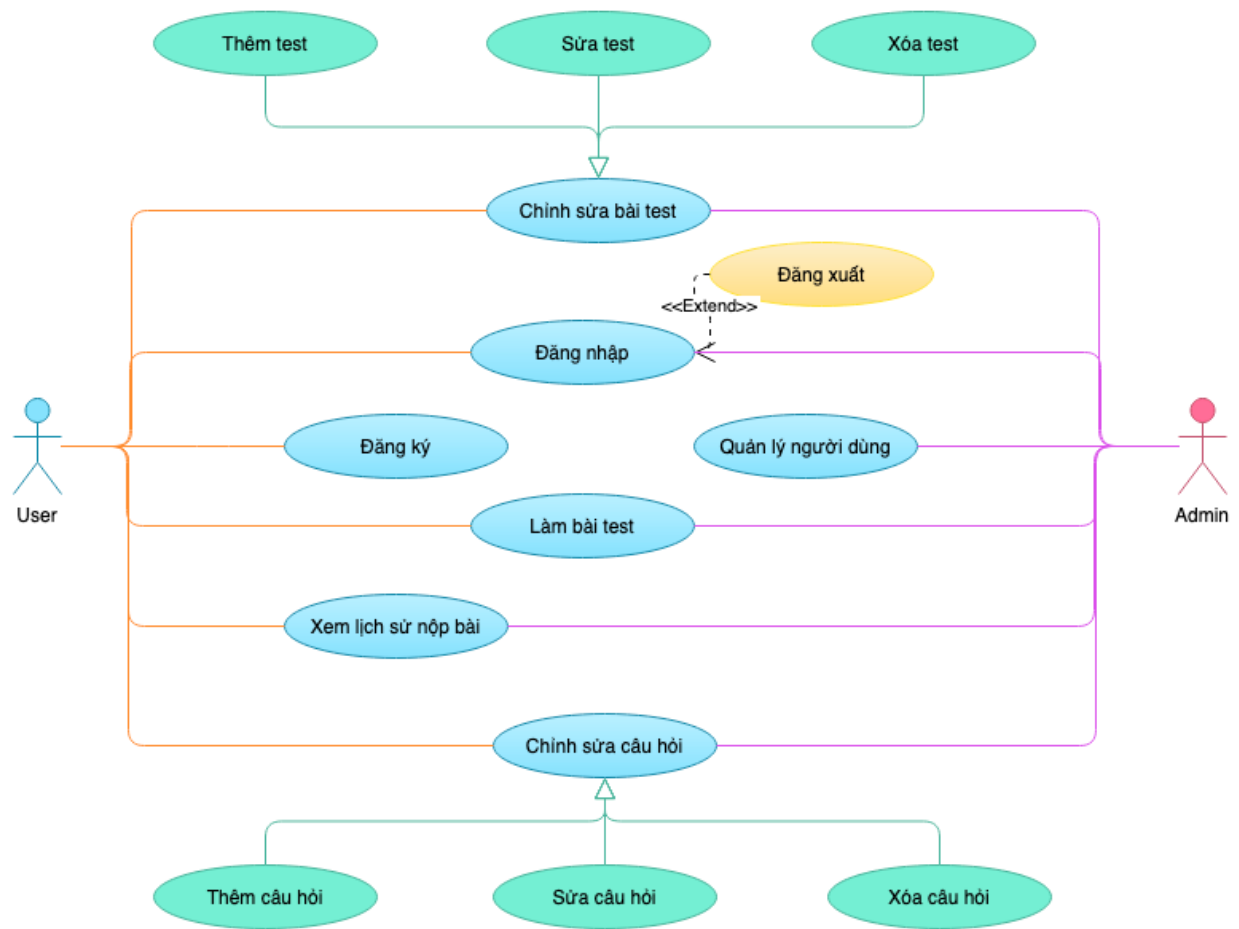
- *Table History*: lưu trữ lịch sử làm bài của User
 - ID (history): kiểu dữ liệu Integer, thuộc tính NOT NULL, AUTO_INCREMENT
 - UserID (ID người dùng): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài User.ID
 - TestID (ID test): kiểu dữ liệu Integer, thuộc tính NOT NULL, khóa ngoài InforTest.ID
 - DateFinish: kiểu dữ liệu timestamp, thuộc tính NOT NULL, Default current_timestamp()
 - Point: kiểu dữ liệu Float, thuộc tính NOT NULL, Default 0

2. Kiến trúc hệ thống

2.1. Architecture



2.2. Use case

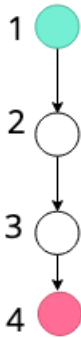


Hình 3.2: Use case diagram User và Admin

Đặc tả use case:

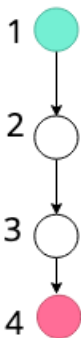
2.2.1. Đăng nhập

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):	<ul style="list-style-type: none"> - Người dùng đăng nhập thành công
<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 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	<ul style="list-style-type: none"> - Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo.
Non-Functional Requirement	<ul style="list-style-type: none"> - Time-out cho màn hình đăng nhập dưới 20 giây

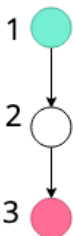
2.2.2. Đăng ký

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):	- Tài khoản người dùng chưa được tạo sẵn

	<ul style="list-style-type: none"> - Thiết bị của người dùng đã được kết nối với database khi thực hiện đăng ký
Post-Condition(s):	<ul style="list-style-type: none"> - 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. Người dùng sẽ được quay trở lại giao diện đăng nhập để thực hiện 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

2.2.3. Đăng xuất

Use Case Name	<ul style="list-style-type: none"> - Đăng xuất
Description	<ul style="list-style-type: none"> - Là người dùng, tôi muốn đăng xuất khỏi tài khoản hiện hành.
Actor(s)	<ul style="list-style-type: none"> - Người dùng, người quản trị hệ thống
Priority	<ul style="list-style-type: none"> - 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:  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) </pre>	<ol style="list-style-type: none"> 1. Người dùng chọn đăng xuất và gửi yêu cầu đăng xuất. 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	- Time out cho màn hình Logout dưới 5 giây.

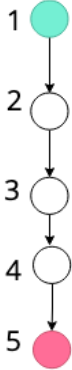
2.2.4. Tạo bài test

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):	<ul style="list-style-type: none"> - 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
Non-Functional Requirement	<ul style="list-style-type: none"> - Time out cho màn hình tạo test dưới 3 giây.

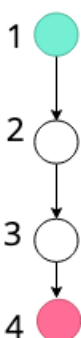
2.2.5. *Chỉnh sửa bài test*

Use Case Name	- Chỉnh 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

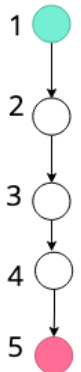
Post-Condition(s):	- Người dùng chỉnh sửa được bài test thành công
Basic Flow:  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 4 --> 5((5)) </pre>	<ol style="list-style-type: none"> 1. Người dùng đăng nhập thành công vào ứng dụng 2. Người dùng chọn bài test muốn chỉnh sửa 3. Hệ thống trả về dữ liệu bài test 4. Người dùng chỉnh sửa bài test 5. Hệ thống cập nhật lại bài test

2.2.6. Xóa 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

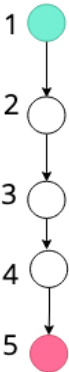
<p>Basic Flow</p>  <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) </pre>	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào ứng dụng 2. Người dùng chọn bài test muốn xóa 3. Hệ thống kiểm tra quyền người dùng 4. Hệ thống xóa bài test trong database
--	--

2.2.7. Thêm câu hỏi vào question store

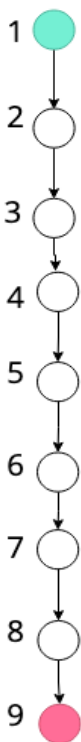
Use Case Name	- Thêm câu hỏi mới vào question store
Description	- Là người dùng, tôi muốn dùng ứng dụng để thêm câu hỏi vào database của mình
Actor(s)	- Người dùng
Priority	- Must Have
Trigger	- Người dùng muốn thêm câu hỏi mới
Pre-Condition(s):	<ul style="list-style-type: none"> - Người dùng đã đăng nhập - Thiết bị được kết nối với database
Post-Condition(s):	- Người dùng thêm 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 điền xong thông tin, ấn Add 5. Hệ thống lưu thông tin câu hỏi vào database và hiển thị vào list câu hỏi

Non-Functional Requirement	- Time out cho màn hình tạo câu hỏi dưới 3 giây.
----------------------------	--

2.2.8. Xóa câu hỏi khỏi question store

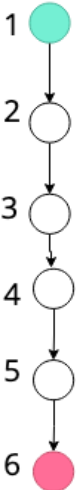
Use Case Name	- Xóa câu hỏi khỏi question store
Description	- Là người dùng, tôi muốn xóa câu hỏi khỏi database của mình
Actor(s)	- Người dùng
Priority	- Must Have
Trigger	- Người dùng muốn xóa câu hỏi cũ
Pre-Condition(s):	- Người dùng đã đăng nhập - Thiết bị được kết nối với database
Post-Condition(s):	- Người dùng xóa 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 hiển 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 hiển thị vào list câu hỏi sau khi đã xóa
Non-Functional Requirement	- Time out cho màn hình xóa câu hỏi dưới 3 giây.

2.2.9. Làm bài test

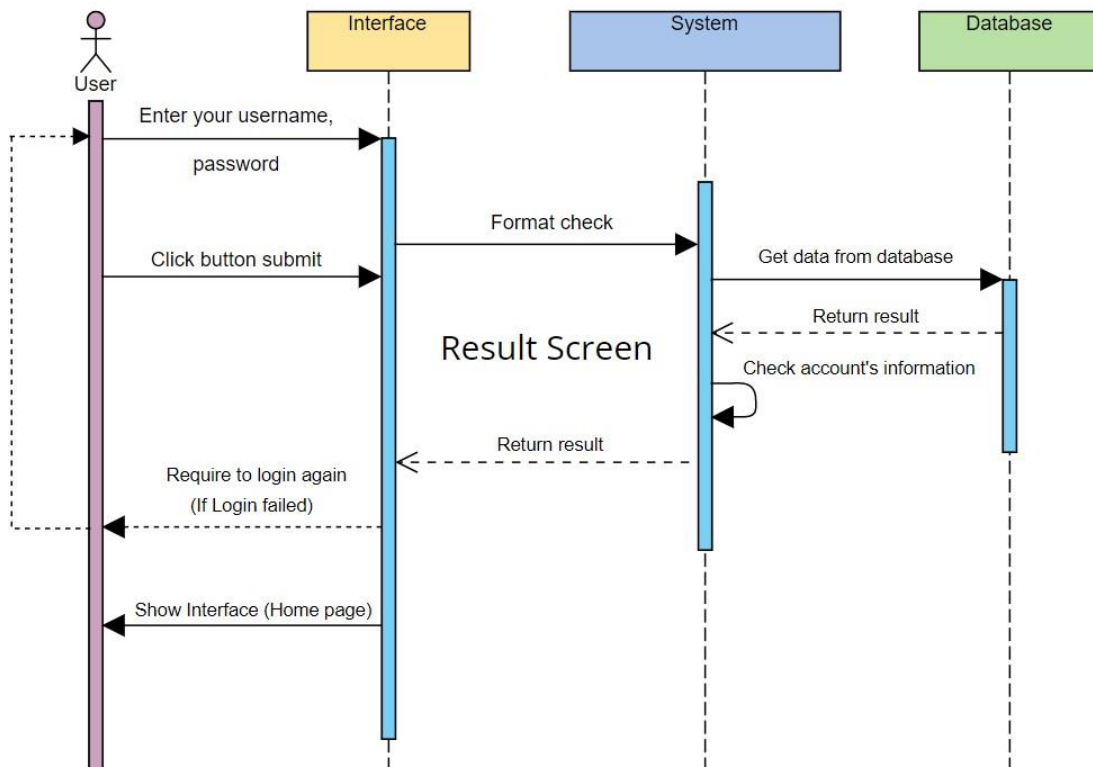
Use Case Name	- Làm bài test
Description	- Là người dùng, tôi muốn tham gia làm các bài test đang hiển thị trên giao diện trang chủ
Actor(s)	- Người dùng, người quản trị hệ thống
Priority	- Must Have
Trigger	- Người dùng muốn tham gia làm các bài test
Pre-Condition(s):	<ul style="list-style-type: none"> - Người dùng đã có tài khoản - 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)) style 1 fill:#00FF00 style 9 fill:#FF00FF </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. Hệ thống chấm điểm bài test và lưu vào database 9. Hệ thống thông báo điểm cho người dùng

Non-Functional Requirement	- Time out cho màn hình submit dưới 60 giây.
----------------------------	--

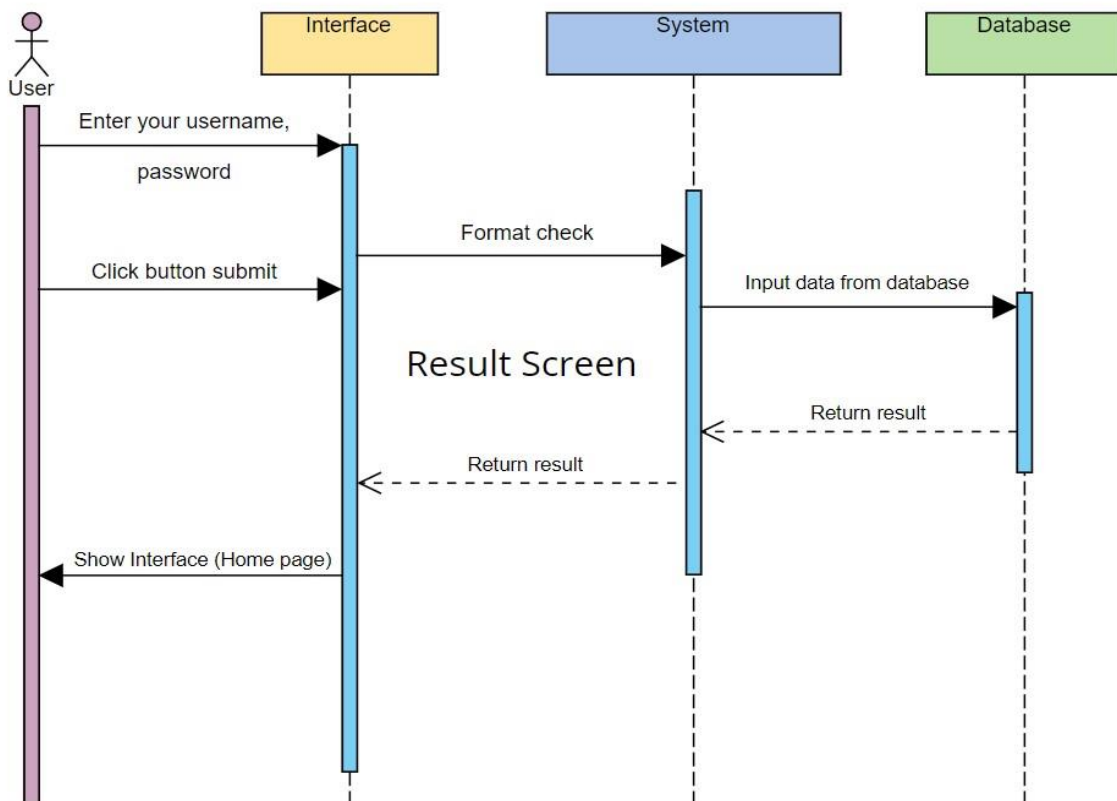
2.2.10. Xem lịch sử nộp bài

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):	- 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 6. Hệ thống load data và hiển thị lịch sử làm bài của người dùng

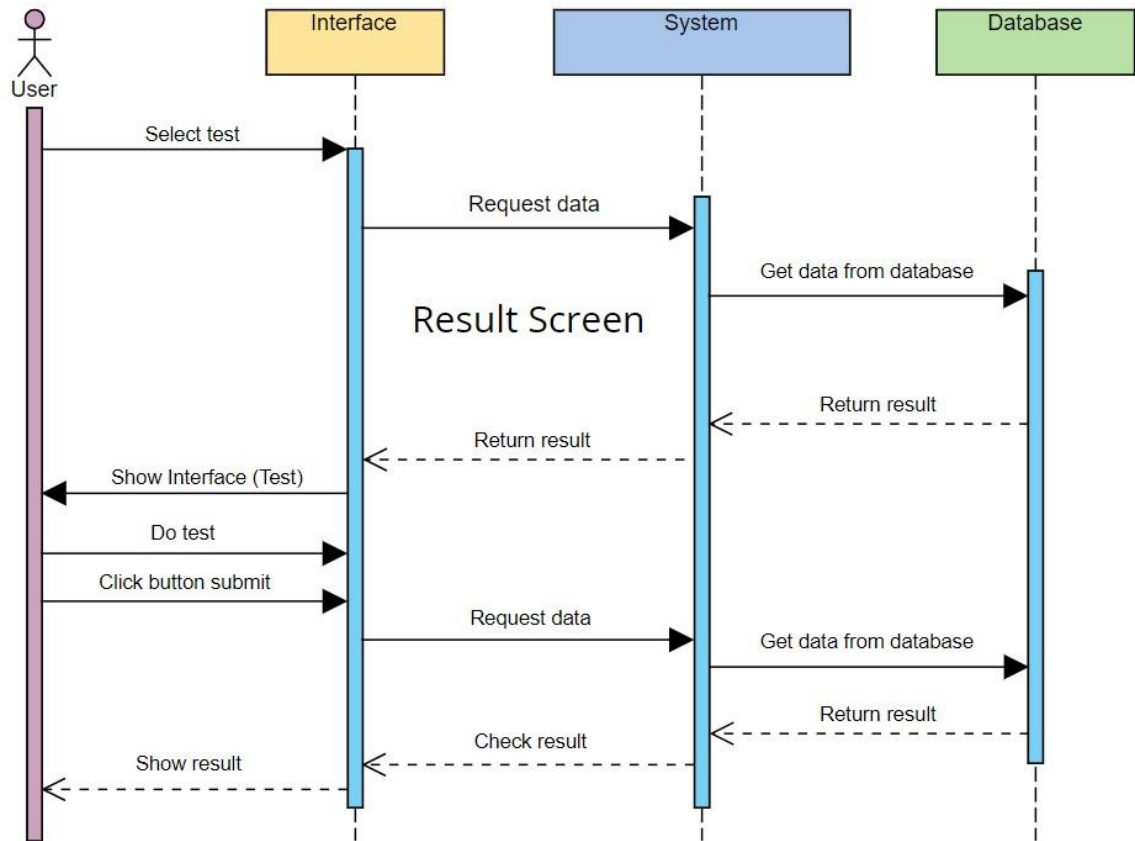
2.3. Sequence diagrams



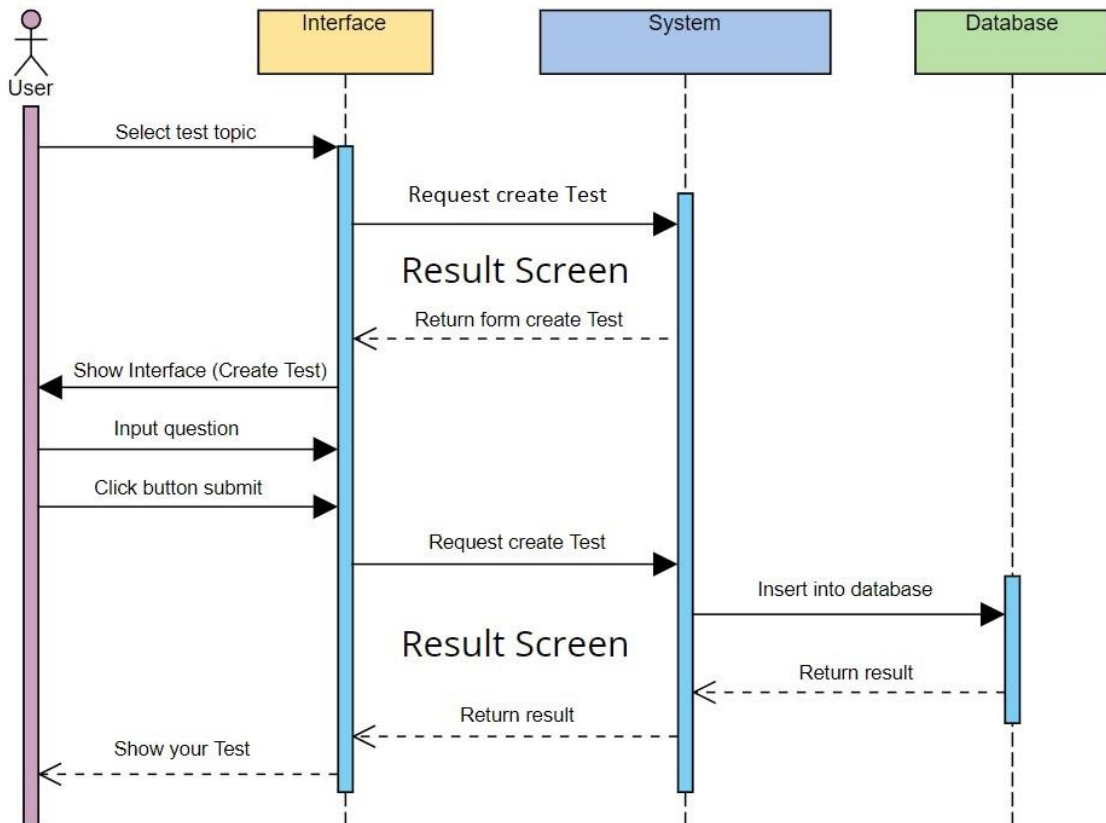
Hình 2.3.1: Sequence diagrams Login



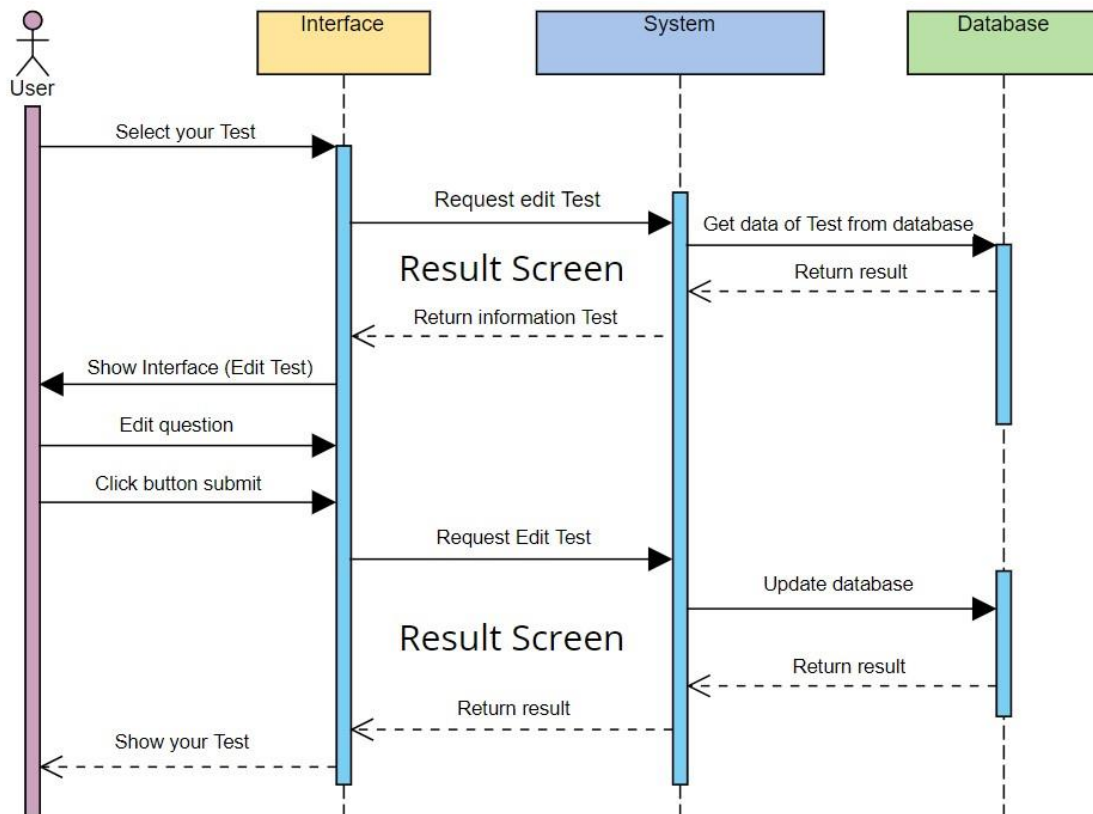
Hình 2.3.2: Sequence diagrams Register



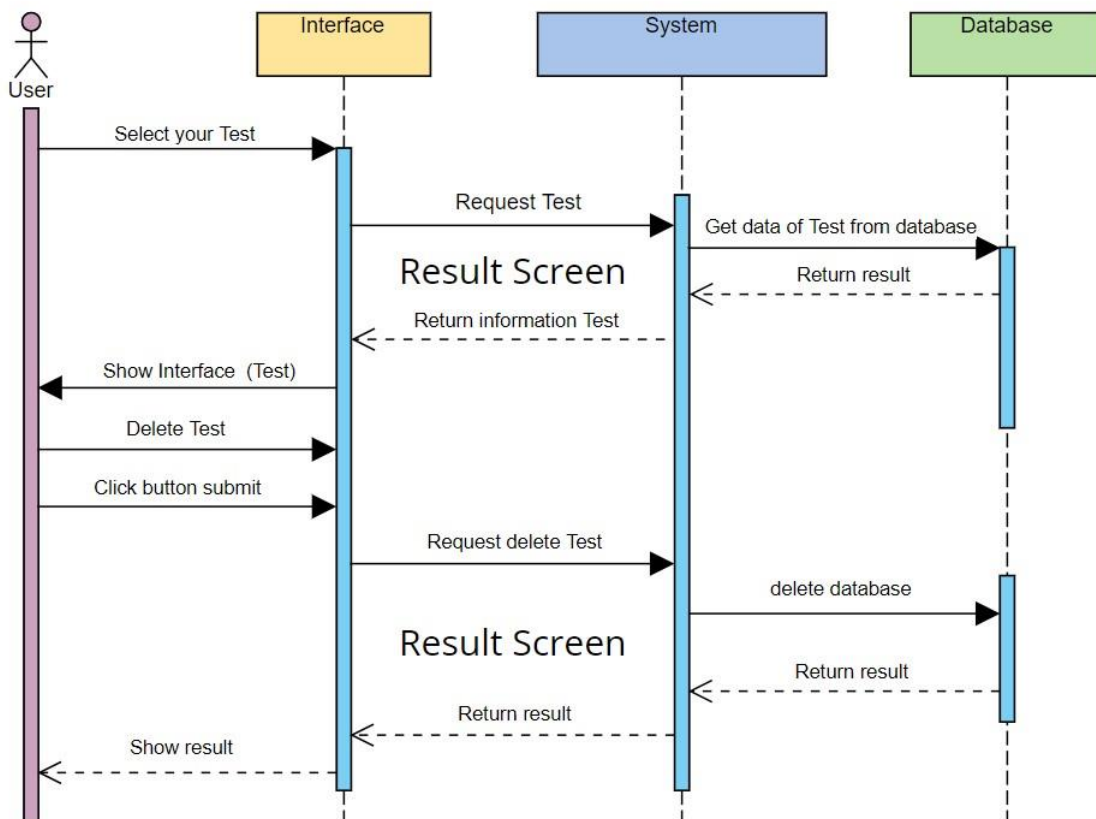
Hình 2.3.3: Sequence diagrams Solve Problems



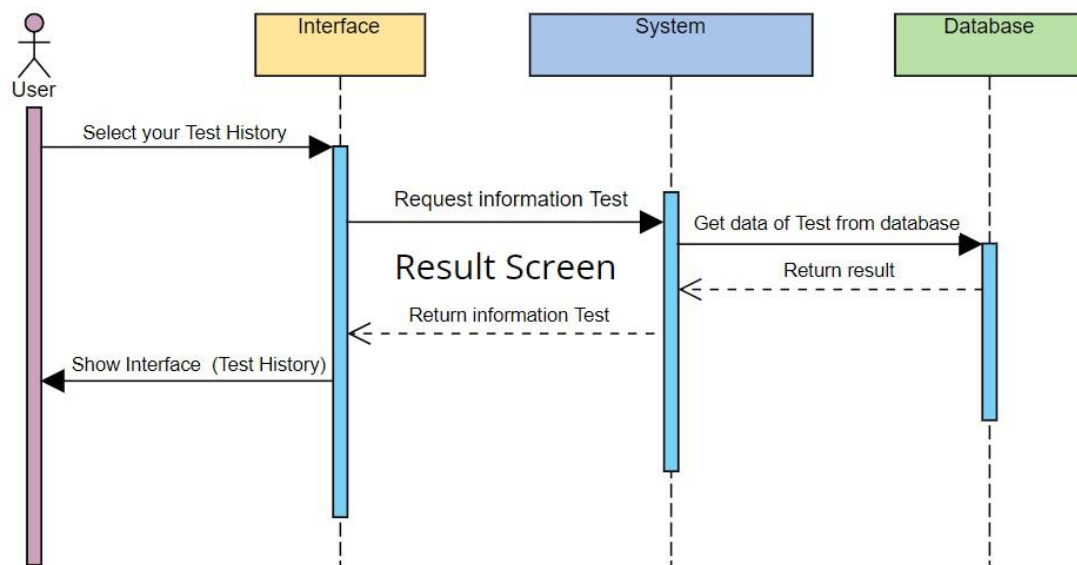
Hình 2.3.4: Sequence diagrams Create Test



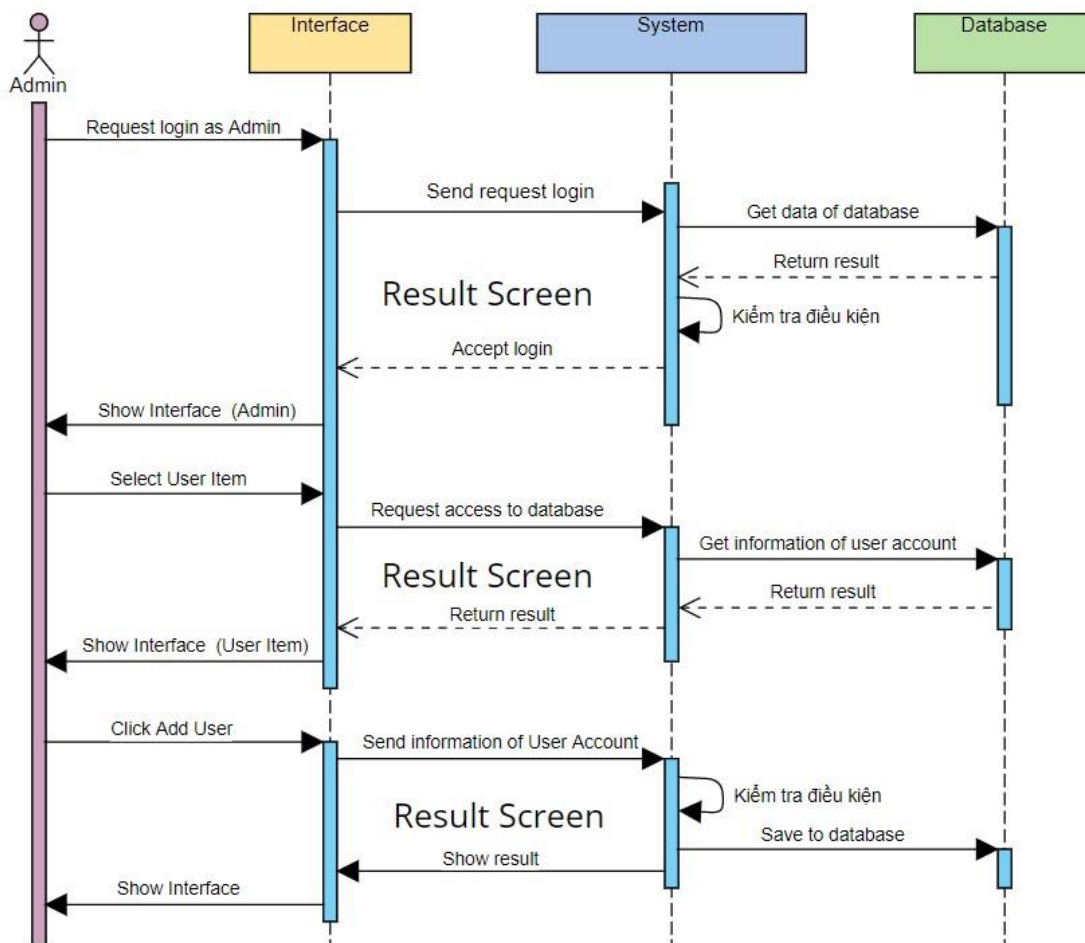
Hình 2.3.5: Sequence diagrams Edit Test



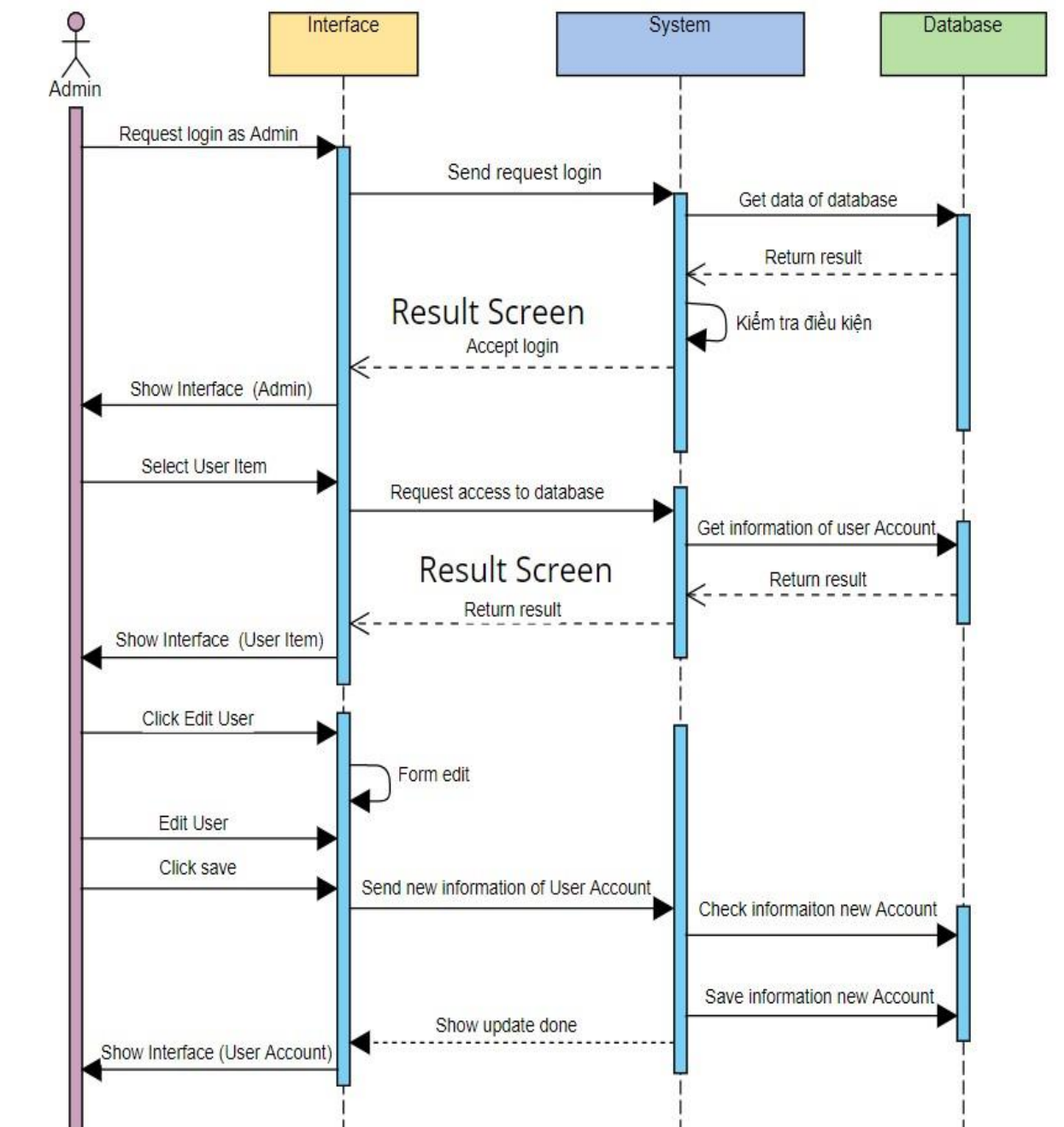
Hình 2.3.6: Sequence diagrams Delete Test



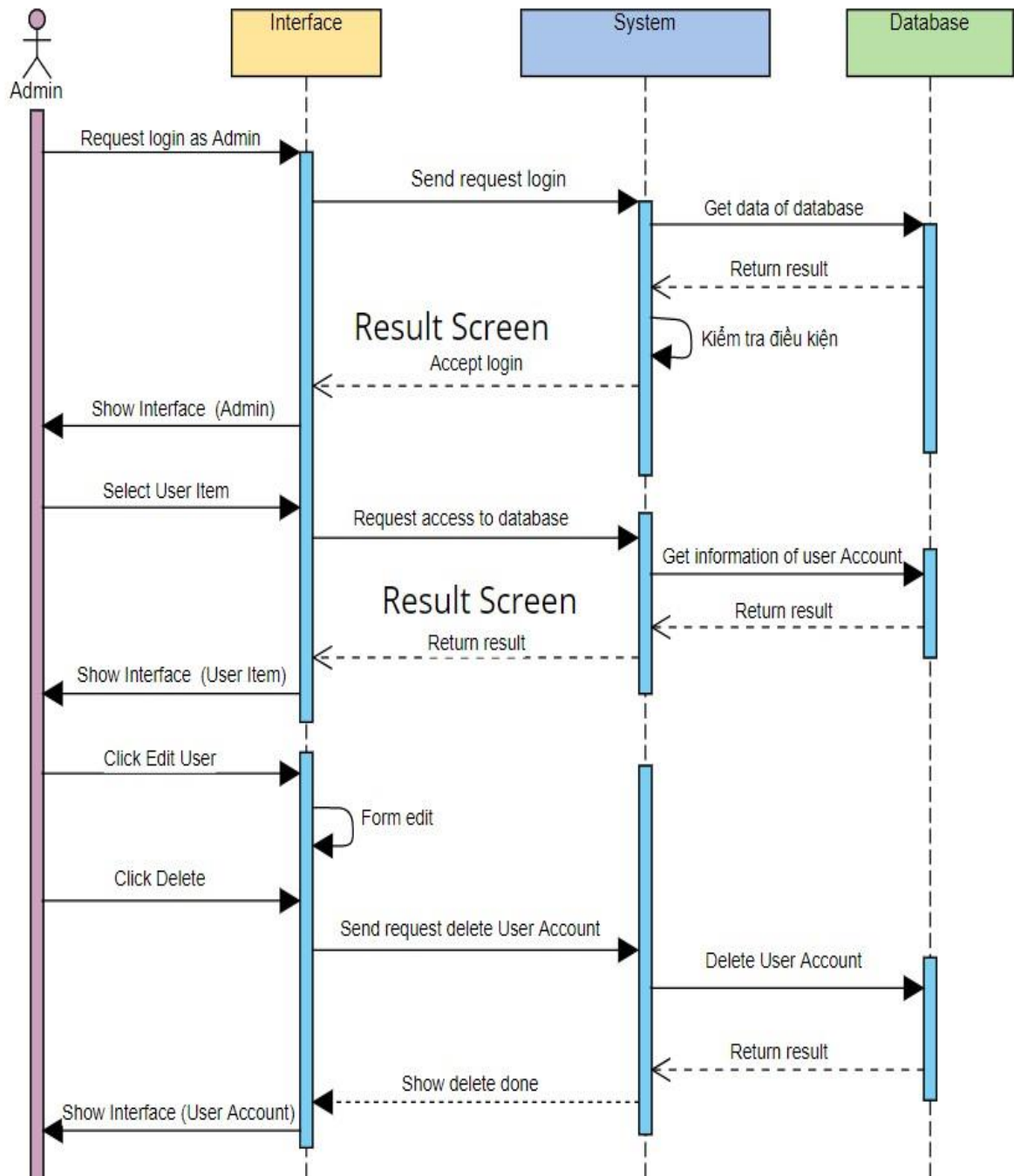
Hình 2.3.7: Sequence diagrams History



Hình 2.3.8: Sequence diagrams Admin Add User



Hình 2.3.9: Sequence diagrams Admin Edit User



Hình 2.3.10: Sequence diagrams Admin Delete User

3. Giao diện hệ thống

3.1. Đăng kí



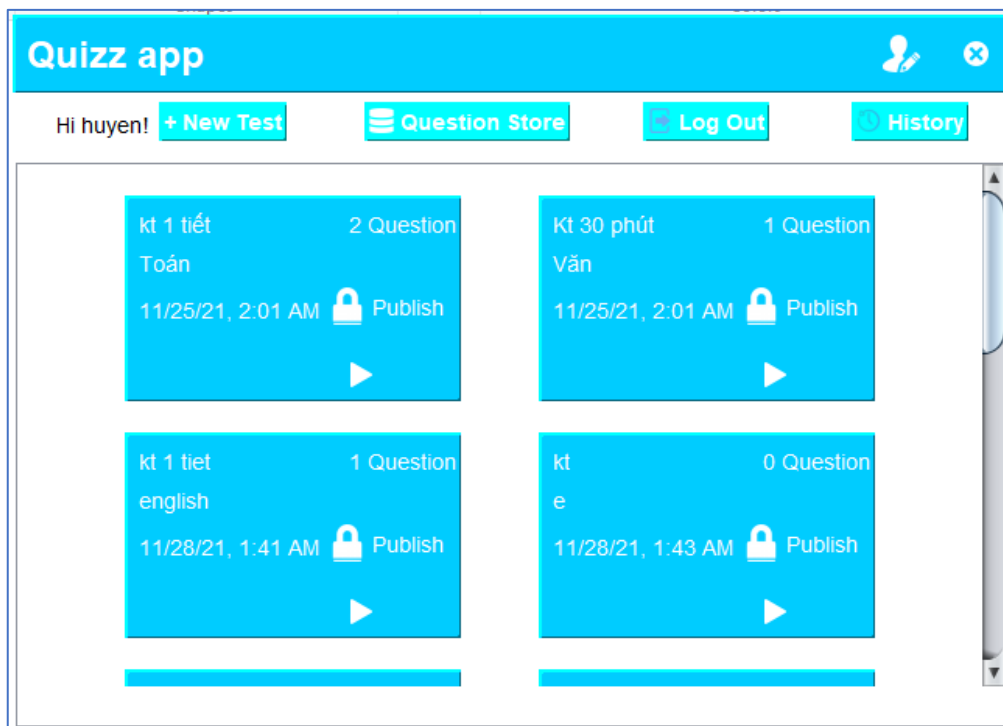
The screenshot shows a mobile application window titled "Quizz App" with standard Android window controls. The main heading is "Đăng ký" (Register) in blue. Below it are three input fields: "Tên đăng nhập" (Username), "Mật khẩu" (Password), and "Nhập lại mật khẩu" (Repeat password). At the bottom, there are two buttons: "Cancel" and "Register".

3.2. Đăng nhập

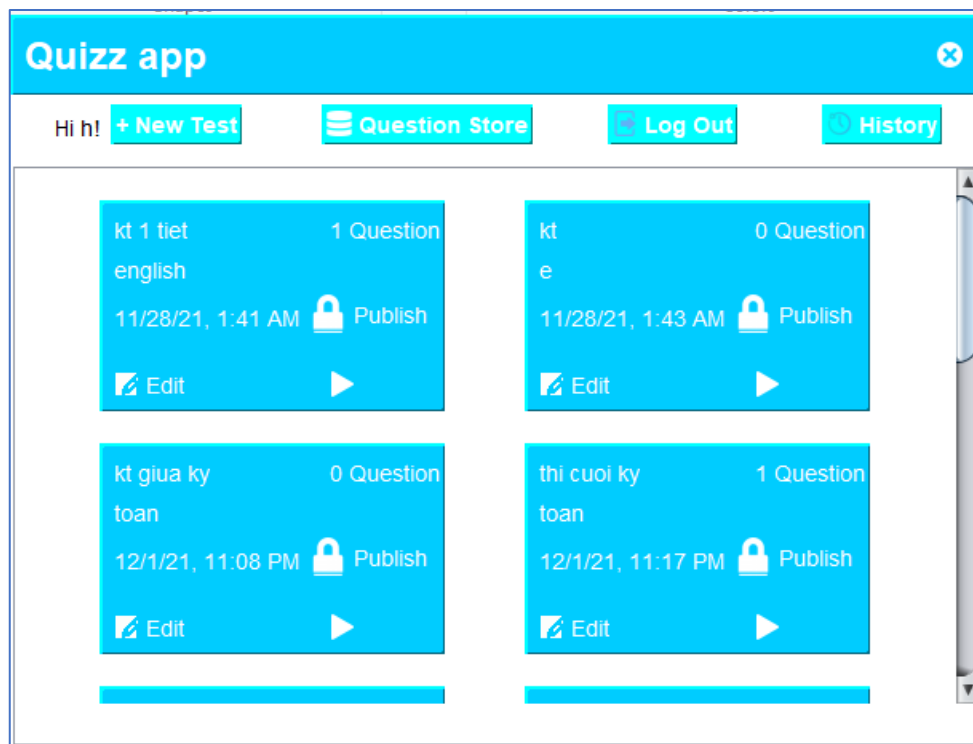


The screenshot shows a mobile application window titled "Quizz App" with standard Android window controls. The main heading is "Đăng nhập" (Login) in blue. Below it are two input fields: "Tên đăng nhập" (Username) and "Mật khẩu" (Password). At the bottom, there are two buttons: "Register" and "Login".



3.3. Trang chủ dành cho Admin



3.4. Trang chủ dành cho người dùng



3.5. Tạo bài test

Quizz app  

Create a new test

Test name




Topic

Publish

☐ Publish ☐ Private

Save

3.6. Thêm/xóa câu hỏi trong kho dữ liệu

 **Question store**  

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



3.7. Lịch sử nộp bài

History

Home

Close

History

Clear All History

Test Name	Topic Name	Number Of Questions	Test Date	Point
-----------	------------	---------------------	-----------	-------

3.8. Chỉnh sửa bài test

The screenshot shows the 'Edit Test' application interface. At the top, there is a blue header bar with the text 'Edit Test' and a home icon. Below the header, the interface is divided into three main sections. On the left, there is a panel titled 'Question In Store' which contains a table with two columns: 'ID' and 'QUESTION'. The table has one row with the value '5' under 'ID' and '4-5' under 'QUESTION'. On the right, there is a panel titled 'Question In Test' which contains a table with two columns: 'ID' and 'CONTENT'. The table has one row with the value '17' under 'ID' and '1-10' under 'CONTENT'. Between these two panels, there are two buttons: a right-pointing arrow button and a left-pointing arrow button. The top of the interface also shows the text 'kt 1 tiet', 'english', and '11/28/21, 1:41 AM'.

3.9. Làm bài test

Doing Test

Test: kt 1 tiet

Topic: toan

Question 1: 1+5

☐ 8

☐ 6

☐ 7

☐ 3

3.10. Quản lý người dùng

Quizz app

STT	User Name	Password	Admin
1	sang	123456a@	<input checked="" type="checkbox"/>
2	bong	1234	<input type="checkbox"/>
3	h	123	<input type="checkbox"/>
4	trang	123	<input type="checkbox"/>
5	nam	12345	<input type="checkbox"/>
6	duonghuyen	123	<input type="checkbox"/>
7	bhhhh	123	<input type="checkbox"/>
8	g	12	<input type="checkbox"/>
9	12	12	<input type="checkbox"/>

Add

Delete

IV. Đánh giá

1. Chương trình đã đáp ứng được những yêu cầu đặt ra
2. Đảm bảo tính dễ dùng
3. Đảm bảo tính nhất quán của dữ liệu
4. Hoạt động ổn định
5. Tốn ít tài nguyên

V. 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
- Hiện thị lịch sử làm bài
- Quản lý người dùng

VI. Đóng góp của từng thành viên

- Nguyễn Hữu Sang:
 - Thiết kế giao diện
 - Code back-end
- Dương Thị Thu Huyền:
 - Thiết kế giao diện
 - Code back-end
- Phan Việt Linh:

- Thiết kế database
 - Thiết kế giao diện
 - Làm báo cáo
- Nguyễn Trung Kiên:
 - Tạo dữ liệu cho database
 - Làm báo cáo
 - Khảo sát và đánh giá
- Nguyễn Thị Thanh:
 - Phân tích yêu cầu
 - Kiểm thử
 - Làm báo cáo

