****

Enterprise Web Software Development

GROUP Report

Term 2 - MAC

Name of the Group: Group 4

* Nguyen Quang Vinh
* Le Thanh Trung
* ....

Greenwich Course Leader: Mr. Matthew Prichard

Class: TCS2006

Subject’s ID: 1640

Assignment due: 16th April 2020

Assignment submitted: 29th

**ASSIGNMENT BRIEF**

|  |  |  |  |
| --- | --- | --- | --- |
| **Degree** | **Honours Diploma in Computing** | | |
| **Unit number** | Term 2 – Level 6 – CW – COMP1640 | | |
| **Assignment title** | Enterprise Web Software Development | | |
| **Academic Year** | 2019 – 2020 | | |
| **Unit Tutor** |  | | |
| **Issue date** |  | **Submission date** |  |
| **IV name and date** |  | | |

|  |
| --- |
| **Submission Format:** |
| *Format:* The submission is in the form of 1 document  You must use font *Calibri size 12, set number of the pages and use multiple line spacing at 1.3. Margins must be: left: 1.25 cm; right: 1 cm; top: 1 cm and bottom: 1 cm.* The reference follows Harvard referencing system.  *Submission:*   * An electronic copy of your work for this coursework should be fully uploaded by midnight (local time) on the Deadline Date. * The last version you upload will be the one that is marked. For this coursework you must submit a single Acrobat PDF document. In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). * For this coursework you must also handling this artefact: Links to repository and screencasts * There are limits on the file size. The current limits are displayed on the coursework submission page on the Intranet * Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions. * Comments on your work will be available from the Coursework page on the Intranet. The grade will be made available in the portal. * You must NOT submit a paper copy of this coursework. * All coursework must be submitted as above   The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences. See http://www2.gre.ac.uk/current-students/regs for details.  *Note:* **Plagiarism** *is presenting somebody else’s work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student’s coursework; stealing or buying coursework from someone else and submitting it as your own work. Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University.*  *All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using. Your work will be submitted for electronic plagiarism checking. Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.* |
| **Deliverables:** |
| **An Individual PDF Report**  The report must give the URL of the Group Repository, the Screencast and the website and any usernames or passwords needed to access it. The individual component of the marking will be based on your report, so ensure this has evidence that your system meets the specified requirements. The text in your individual report must be entirely your own words. |
| **Assignment Brief and Guidance:** |
| **Scenario**:  This is a group coursework with a maximum of six in the group.  You need to adopt agile scrum working practices, and document your meetings appropriately. Ideally you need a database designer, a programmer, a web designer and a tester, but you should take on all these roles at various stages as part of the project, and more than one person can be in any role at any time. No one is to take the role of project manager.  You will get an individual grade based on your contribution to the team, and for your individual contribution to the product.  **Specification:**  You are required to build a web-based secure role-based system for eTutoring in a large university. Full details of the system will be given in lectures.  The system must meet the following criteria:  • All students must have a personal tutor.  • Any authorized member of staff can allocate or reallocate personal tutors to students. The student and the personal tutors will get notification emails when this happens.  • Bulk allocation of students to their personal tutor (eg 10 at a time) needs to be implemented.  • All students and their tutors are to use the eTutor system for messaging, arranging and recording meetings (both real and virtual), uploading documents and commenting on them, and for blogging.  • Email between students and their personal tutors is to be used only for notification of events recorded in the backend database. No other content is to be sent via email.  • Student and staff data is accessed from the university MIS system. The maintenance of this is outside the scope of this project.  • Each student will have their own personal dashboard summarizing their interaction with their personal tutor.  • Each personal tutor will have a dashboard of their personal tutees that can be sorted and filtered appropriately  • Authorized staff will have access to the dashboards of other staff, and to individual dashboards for students.  • The interface must be suitable for all devices (eg mobile phones, tablets, desktops)  **Assumptions:**  You must clearly state any assumptions you make.  **Reports:**  A number of reports need to be made available. For example:  • Statistics  o Number of messages in last 7 days  o Average number of messages for each personal tutor  • Exception reports  o Students without a personal tutor.  o Students with no interaction for 7 days and 28 days.  **Tasks:**  1. Work as a team using agile scrum methods to develop and test a secure web-based system to meet the above specification.  2. Create a screencast recording (including screen and sound) demonstrating the key functionalities of the system. This needs to be hosted somewhere (e.g. YouTube) that is accessible by the Greenwich moderator  3. Present the finished product to a non-technical audience to try to persuade them to purchase your system.  4. Document the system to an appropriate standard using a weighted scoring model with commentary, including an evaluation of the design process you followed and your reflection on the finished product, and on the contributions of your team members. |

|  |  |  |
| --- | --- | --- |
| **Assessment Breakdown** | | |
| Group Component | **Total:** 60**%** | This will be assessed based on a group report and a group repository created by the group on a secure shared area accessible to the Greenwich markers. Password and URL must be provided in individual reports. Must be suitably structured with a menu. Suggested location: GitHUB, Google Docs, SharePoint 365, own website, DropBox or other repository. |
| Database | 10% | Expect: Security, appropriate data types and validation, clear ERD, referential integrity implemented, enables roles to be implemented |
| Site design | 10% | Expect: Responsive design, clear information architecture for both mobile and desktop, aesthetically pleasing, good usability, meets accessibility criteria |
| Functionality | 10% | Expect: Role based security, submission of reports, email notification, summary and exception reports, UML diagrams, code snippets |
| Testing | 10% | Expect: Test plan, test log, sufficient data to fully test, evidence of testing finding errors, test items linked to user stories in the product backlog |
| Agile methods followed | 10% | Expect: Burn down chart, minutes of meetings, user stories, sprints, product backlogs |
| Screencast and Presentation | 10% | Expect: Professional standard of presentation promoting the product, with contributions by all the team members, Screencast demonstrating all the main features of the product. Screencast can be done by one person. |

|  |  |
| --- | --- |
| **Indicative Grading Criteria** | |
| >=70% | * Well designed system to fully meet the requirements. * Professional standard of report, with appropriate documentation. * High level of individual commitment. * High level of evaluative commentary |
| 60-69% | * Well designed system to meet most of the requirements * Professional standard of report * High level of individual commitment * Limited evaluative commentary |
| 50-59% | * Well designed system to meet most of the requirements * Acceptable standard of report * Good level of individual commitment * Limited evaluative commentary |
| 40-49% | * Acceptable system to meet most of the requirements * Acceptable standard of report * Acceptable level of individual commitment * Limited evaluative commentary |
| <40% | * Poorly designed system * Few requirements met * Poor standard of report * Limited individual commitment * No evaluative commentary |

Contents

[**SCHEDULE** 7](#_Toc33889123)

[**AGILE METHODS FOLLOWED** 7](#_Toc33889124)

[*Burn down chart* 7](#_Toc33889125)

[*Minutes of Meetings* 7](#_Toc33889126)

[*User stories* 7](#_Toc33889127)

[*Sprints* 7](#_Toc33889128)

[*Product backlogs* 7](#_Toc33889129)

[**DATABASE** 8](#_Toc33889130)

[*SECURITY* 8](#_Toc33889131)

[Pros: 8](#_Toc33889132)

[Cons: 8](#_Toc33889133)

[*Validation & Datatypes* 8](#_Toc33889134)

[*ERD*: 8](#_Toc33889135)

[*Database Explaination:* 8](#_Toc33889136)

[**SITE DESIGN** 8](#_Toc33889137)

[*What is Responsive Design* 8](#_Toc33889138)

[*How do we apply it to the project* 8](#_Toc33889139)

[*Interface Design* 8](#_Toc33889140)

[*How is this work* 8](#_Toc33889141)

[**FUNCTIONALITY** 9](#_Toc33889142)

[*The application of Role base system* 9](#_Toc33889143)

[*The Report from the system* 9](#_Toc33889144)

[*The Email from the system* 9](#_Toc33889145)

[*UML diagram* 9](#_Toc33889146)

[**TESTING** 10](#_Toc33889147)

[*Test Plan* 10](#_Toc33889148)

[*Test log* 10](#_Toc33889149)

[*ERROR* 10](#_Toc33889150)

[**SCREENCAST & PRESENTATION** 10](#_Toc33889151)

[**LOCATION REPOSITORY** 10](#_Toc33889152)

[**REFERENCES** 10](#_Toc33889153)

# **SCHEDULE**

* Kế hoạch phát triển ban đầu sẽ theo hướng nào, tại sao
* Nêu ra: có những tasks nào cần thực hiện, liên quan như thế nào đến đề bài
* Phân công các thành viên trong nhóm vào các vị trí : Database design, web design, programmer, tester – mỗi thành viên phải carry ít nhất 3 roles khác nhau (đổi role khi qua sprints mới)
* Meeting khi nào, dự kiến ở đâu, mục đích của các cuộc meeting đó là gì

# **AGILE METHODS FOLLOWED**

* Hình minh họa
* Định nghĩa sơ về Agile-scrum

## *Burn down chart*

* Định nghĩa sơ về Burndown chart
* Hình burndown chart của project hiện tại + đưa ra 1 file riêng để gửi riêng cho trường
* Giải thích ý nghĩa từng cột
* Giải thích một số số liệu bất thường (ví dụ ngày 12 thì số lượng công việc hoàn thành ít, do có thành viên bị ốm)

## *Minutes of Meetings*

* Định nghĩa, giải thích sơ minutes of meetings trong scrum
* Toàn bộ report của minutes of meeting (table) + đưa ra 1 file riêng để gửi riêng cho trường

## *User stories*

* Định nghĩa sơ user stories trong scrum
* Toàn bộ user stories trong dự án (dưới dạng table)

## *Sprints*

* Định nghĩa sơ sprints trong scrum
* Toàn bộ các sprints đã chạy trong dự án (dưới dạng table), sprint backlog + đưa ra 1 file riêng để gửi riêng cho trường
* Dưới mỗi table phải giải thích tại sao có sprints này (ví dụ, Login là function quan trọng then chốt, nên phải có sprints làm Login riêng)

## *Product backlogs*

* Định nghĩa sơ product backlogs trong scrum
* Product backlogs của dự án + đưa ra 1 file riêng để gửi riêng cho trường
* Giải thích tại sao có product backlog này (ví dụ: khách hàng yêu cầu Student phải có dashboard)

# **DATABASE**

## *SECURITY*

### Pros:

* Giai thich thuan loi trong viec tao database nhu the nay (ví dụ: truy xuất nhanh)
* Hinh anh minh hoa 1

### Cons:

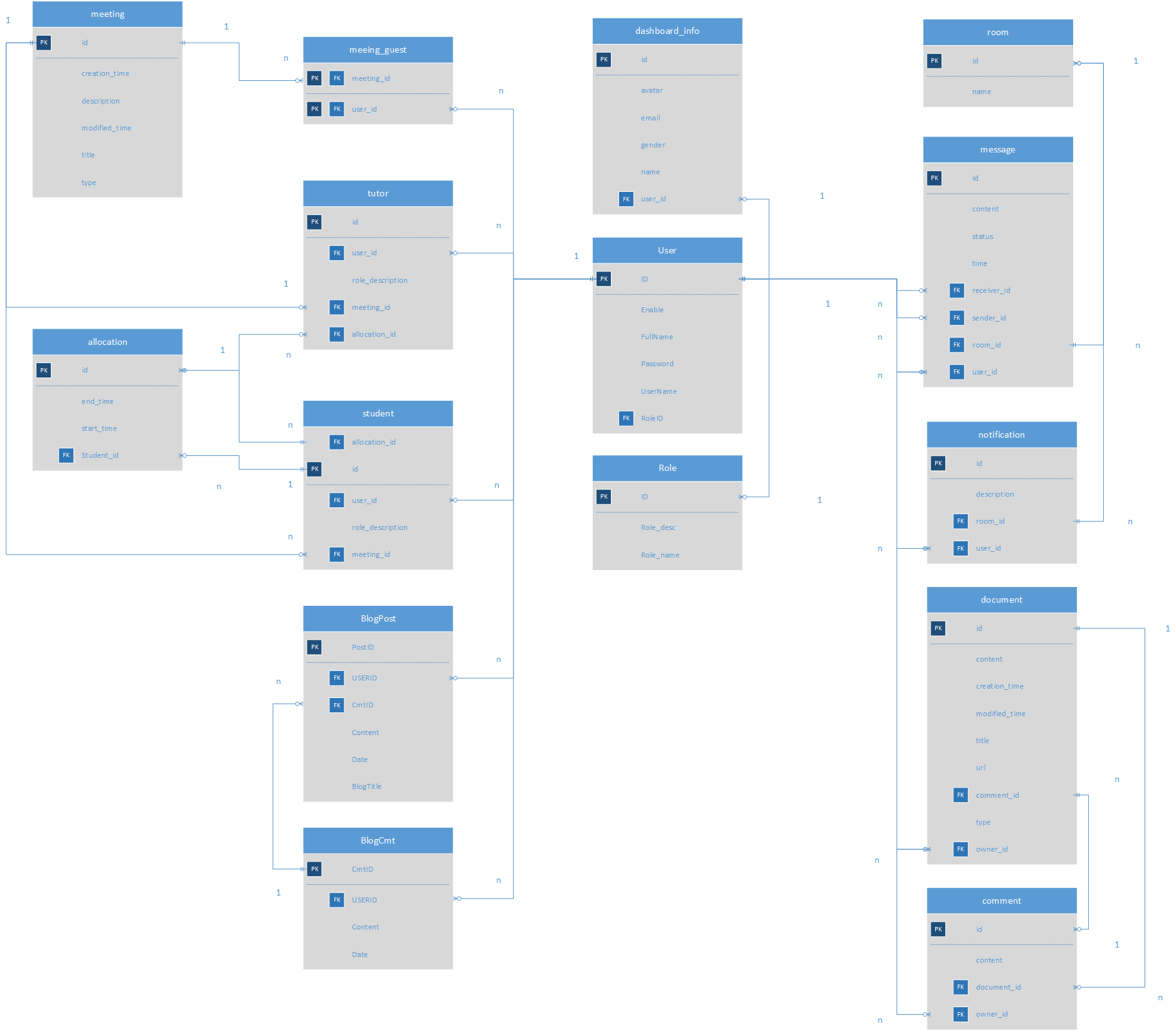
* Neu mot so bat loi trong viec tao database nhu the nay (vi du: lap du lieu...)
* Hinh anh minh hoa 2

## *Validation & Datatypes*

* Neu mot so datatype được dùng trong CSDL
* Giải thích tại sao dùng datatype đó
* Validation như nào

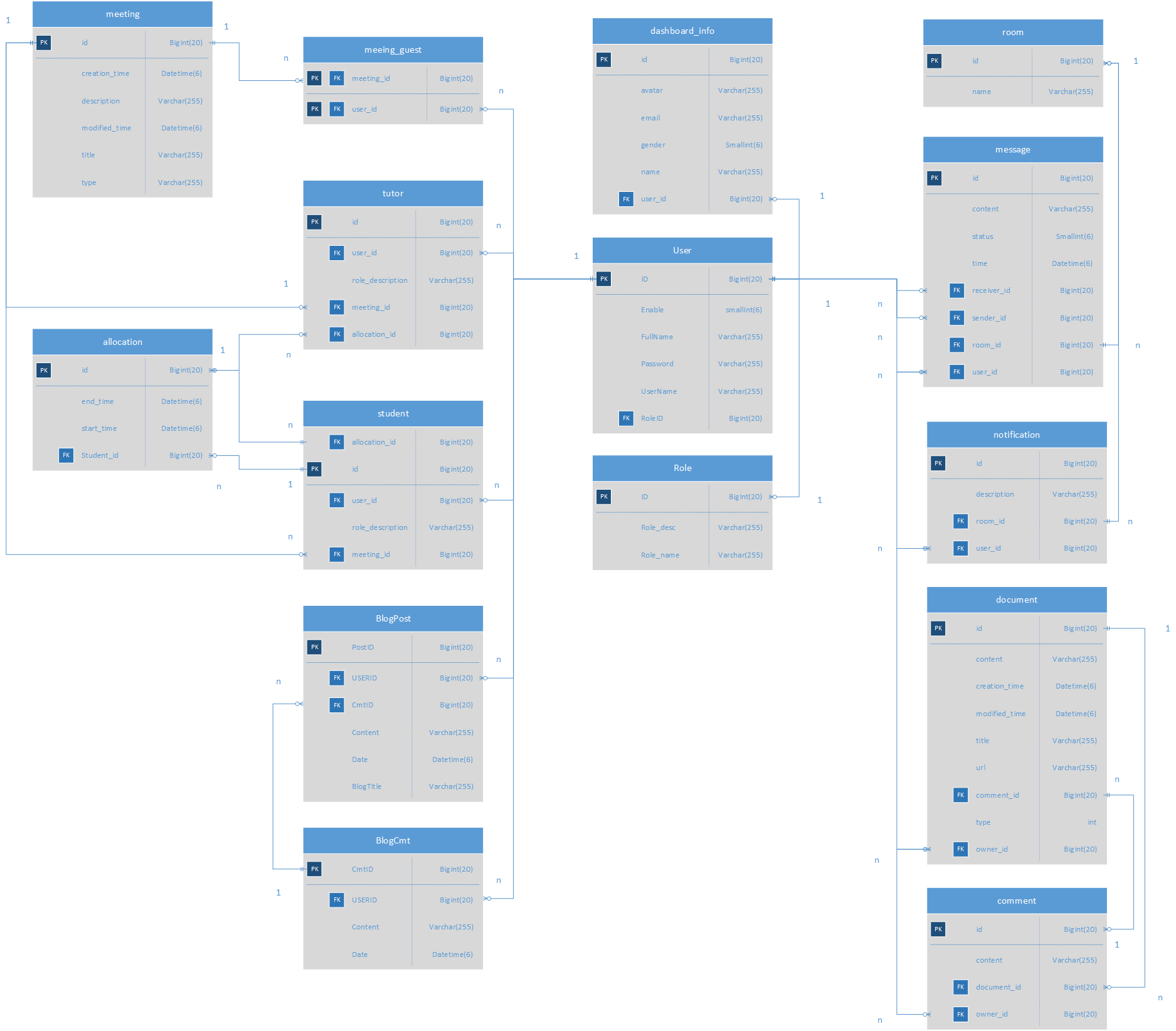
## *ERD*:

Entity relationship Diagram (without datatype version)



There is the ERD of the database that will be used on this project, but this one is the one without the datatype. It shows really clearly the relationship between each table. Although the picture is a quite small, please check the explain below for more information when we dig into the database one-by-one function.

Entity relationship Diagram (with datatype version)



Another ERD for the database, but this is the version with datatype and its limit. The picture is quite small when the database itself is quite large, so please check the explain section below for more information.

## *Database Explaination:*

* Giải thích các mối quan hệ trong CSDL (chụp ảnh từng bảng có quan hệ để giải thích)

Role base function

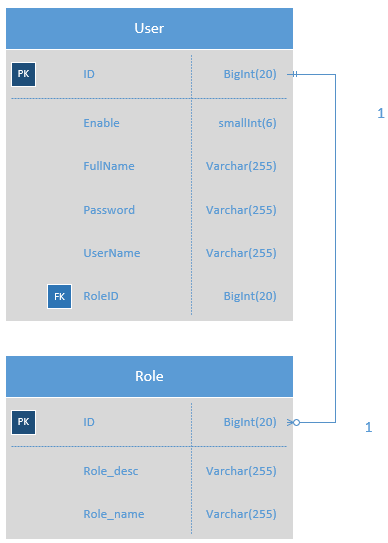
****

Figure : Role base function

When user using login function, the table User and Role will be in used. Besides having different UserID, each user also has RoleID to identify which level of authority they have, such as being a Staff, or a Student, or a Tutor ... will be determined by the RoleID. Please keep in mind that the RoleID field may or may not have the same name as the role of user in real life, for example: the RoleID can be 1, but the real-life role of user is Staff.

Since the delete function is necessary for the user, the field “Enable” has been put in, each and every time user log in, the system will check if this field is valid to see the status of the account. Why do we just use the delete function on database, you ask? Simply because this is a relationship database, so delete 1 data may cause the huge problem for the other data, because they have “relationship” with each other, so the best way is to keep it as it is, and then just “hide” them.

Message function

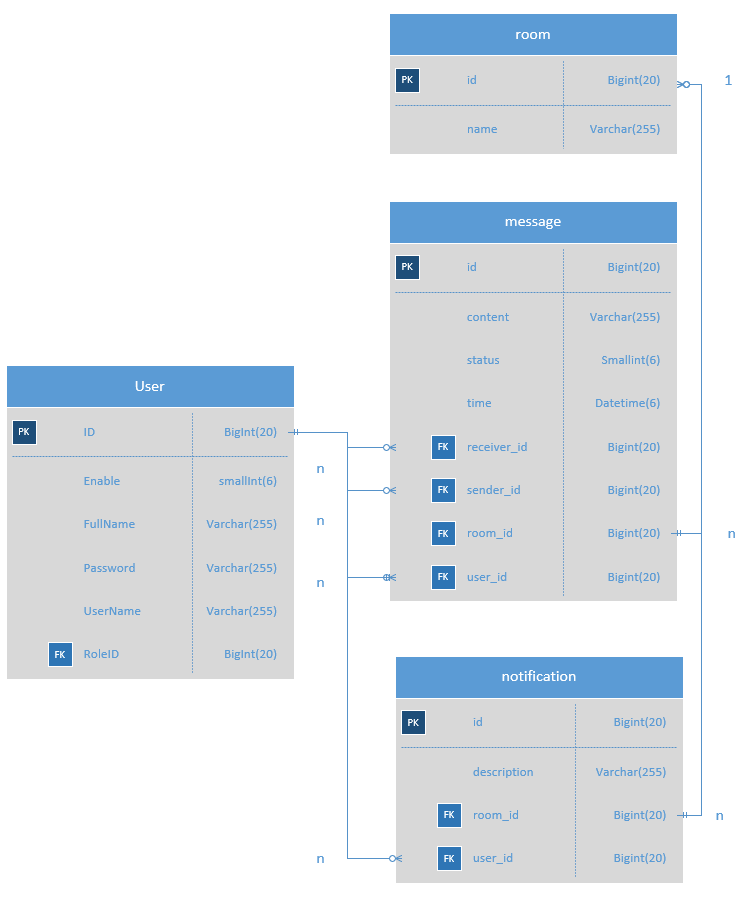


Figure : Message function

When the user enters Message Page, the User and Message table will be used. Each User already have an identical UserID, and the SenderID and ReceiverID from Message table will use this characteristic to identify who will be the sender and who is the receiver to help build the Notification and Message functions of the application.

The room table will cover the whole things including both message and notification table, and make it easier to build the front-end code. Every time a message has been sent to the room, we can identify right away who is the owner, and then the system will send the notification to all other members in the room except the sender of the message.

Also, we can identify which notification we get come from what room, and what message will be sent to what room, all has been covered.

Allocating function

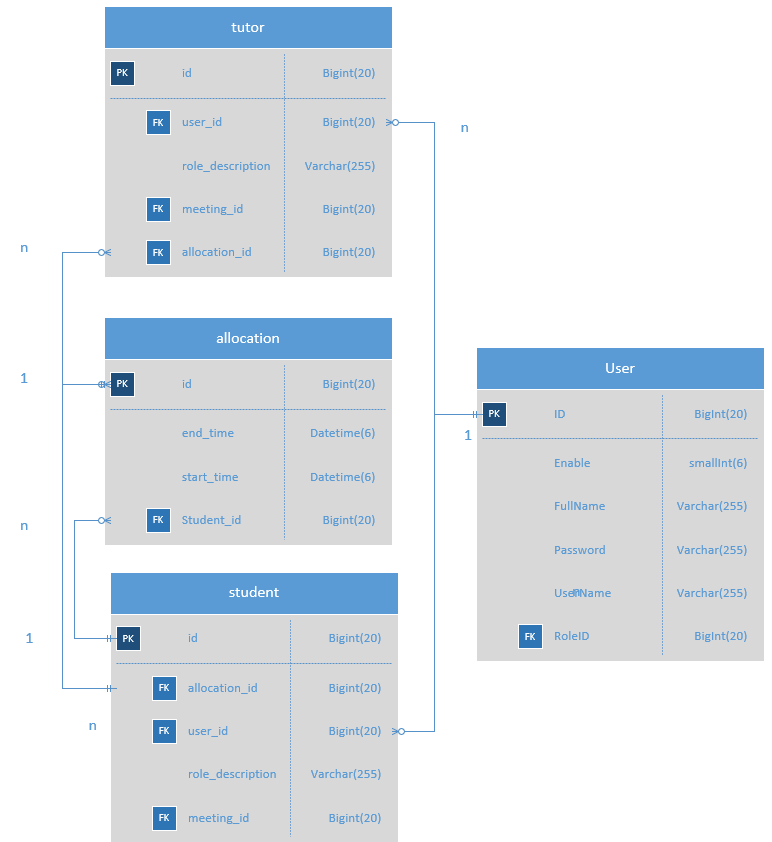


Figure : Allocating function

It’s a bit complex but with 4 tables: User, Tutor, Student, and Allocation, that will be come easier to build the allocating function. With the Tutor and Student table, the Allocation table will get the right ID for it locating function, whose will be assigned with whom.

Because the relationship in this case somehow is many-many between User and Allocating table, so that we will need to break it down to 1-many and many-1 tables to help present the data better.

For each “allocation” we only have 1 tutor ID but may have many student ID, and for each tutor, they can be allocating to many other “allocation”. For better understanding, we can think it as a “Classroom”, only 1 Tutor for each “Classroom” with many “Students” at a time, and then when its time, the Tutor can switch to other “Classroom” with different “Students”.

Meeting function

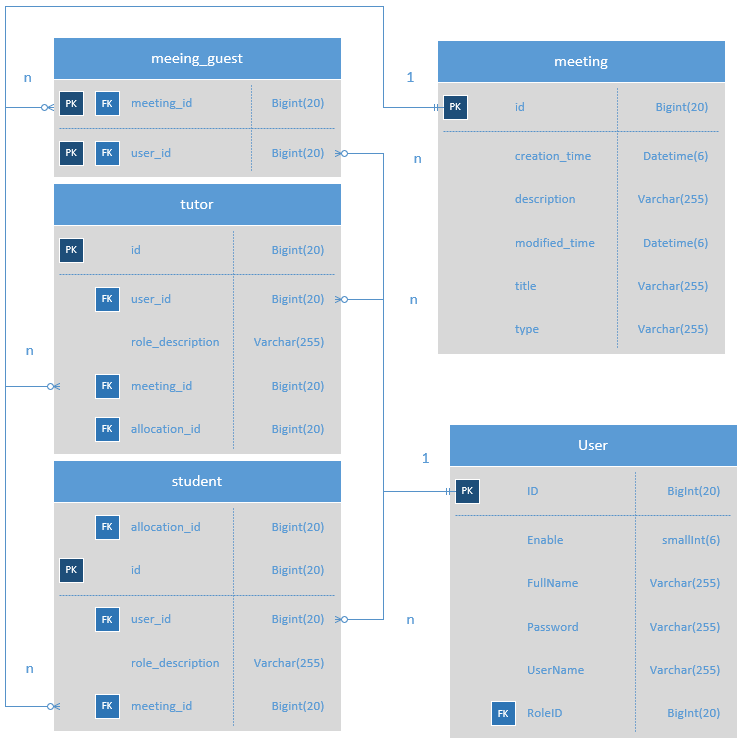


Figure : Meeting function

Meeting function in this case will use the same output of Tutor and Student table just as Allocating function, since the concept is the same, holding a “Classroom” with 1 Tutor at a time

Document and Comment Function

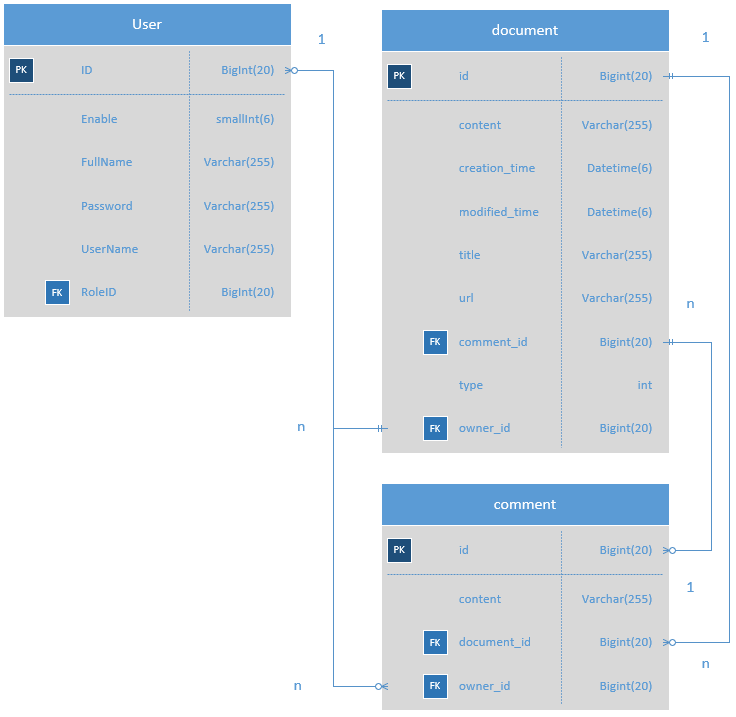


Figure : Upload funtion

One of the important functions is upload file. Sometime user will need to upload the file to the internet, and then the other users will come, check the file, and see if there is any problems, and then leave a comment beside this document and expected the change from it’s owner. The OwnerID will be determined through UserID from User table, and also

Blog function



Figure : Blog function

# **SITE DESIGN**

## *What is Responsive Design*

* Định nghĩa sơ Responsive Design

## *How do we apply it to the project*

* Hình chụp code minh họa
* Giải thích mình đã dùng những gì trong frontend để nó trở thành Responsive Design

## *Interface Design*

* Chụp ảnh toàn bộ wireframe của các trang lên (máy tính)
* Chụp ảnh toàn bộ wireframe của các trang lên (điện thoại)

## *How is this work*

* Hình full screen chạy trên máy tính
* Giải thích những function nào sẽ được hiển thị, function nào ẩn đi bớt, có thân thiện với người dùng không, vì sao
* Hình bị kéo nhỏ trên máy tính
* Giải thích những function nào sẽ được hiển thị, function nào ẩn đi bớt, có thân thiện với người dùng không, vì sao
* Hình trên điện thoại
* Giải thích những function nào sẽ được hiển thị, function nào ẩn đi bớt, có thân thiện với người dùng không, vì sao

# **FUNCTIONALITY**

## *The application of Role base system*

* Giải thích tại sao chia role lại tốt cho việc bảo mật
* Hình vài đoạn code minh họa cho việc chia role sẽ dễ phân loại người dùng hơn

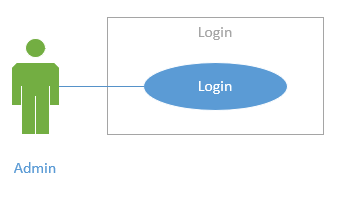
## *The Report from the system*

* Hình một cái report hoàn chỉnh về Student được xuất ra từ hệ thống
* Giải thích vài trường quan trọng trong cái report
* Hình một cái report hoàn chỉnh về Student được xuất ra từ hệ thống (Exception report)
* Giải thích vài trường quan trọng trong cái report, highlight tại sao report này lại khác report trên

## *The Email from the system*

* Giải thích lại yêu cầu của đề bài về email
* Hình một đoạn code minh họa về gửi notification khi có email
* Giải thích đoạn code đó
* Hình về một cái notification khi có email trên trang web
* Giải thích, đánh giá cái notification đó có ý nghĩa thế nào với tutor/student

## *UML diagram*

* Giải thích sơ UML class diagram là gì
* Hình Class diagram của cả dự án.
* Giải thích hướng đi của một số luồng data chính (ví dụ như Login dưới danh nghĩa Staff để alocating student và tutor)
* Hình Use Case của từng staff/tutor/teacher ví dụ
* 
* Giải thích từng key components cho mỗi use case (name, actors, description, goal, pre-condition, trigger, post-condition, normal flow, alternative flow, exception flow) ví dụ

|  |  |
| --- | --- |
| **Key components** | **Explanation** |
| **Name** | Login |
| **Actors** | Admin |
| **Description** | Login into the specific manager page |
| **Goal** | Reach the manager page of Admin |
| **Pre-condition** | Main Page is displayed successfully |
| **Trigger** | User press Enter or Login button |
| **Post-condition** | Success page appears, user has been redirected to manager page of Admin |
| **Normal flow** | [User logged in]  1. User reach main page  2. User input username/password  3. User press Enter or Login button  4. User reach Success page  5. User has been redirected to Manager page of Admin |
| **Alternative flow** | [Wrong Username/Password]  Show the Failed page and redirect user back to main page  Repeat steps in normal flow |
| **Exception flow** | [Special character on Username/Password]  Show the error page  Return the main page again |

* Frequence Diagram của toàn bộ hệ thống

# **TESTING**

## *Test Plan*

* Định nghĩa sơ test plan là gì
* Toàn bộ test plan (dưới dạng table) + user stories của mỗi phần test

## *Test log*

* Toàn bộ test log (dưới dạng table) + đưa ra 1 file riêng để gửi riêng cho trường

## *ERROR*

* Hình một số phần test bị fail (frontend)
* Các test bị fail (dưới dạng table)
* Hướng giải quyết các test fail

# **SCREENCAST & PRESENTATION**

* Link Slide giới thiệu sơ về toàn bộ dự án
* Link Đoạn quay phim đang presentation của toàn bộ các thành viên (từng người 1, cắt ghép video) + đưa ra 1 file riêng để gửi riêng cho trường
* User Guide cho toàn bộ các function chính của sản phẩm (hướng dẫn dùng bằng hình ảnh) + đưa ra 1 file riêng để gửi riêng cho trường

# **LOCATION REPOSITORY**

* Link để download toàn bộ bài

# **REFERENCES**

Phần này cái nào ở trên có định nghĩa thì mới có references – 10 references ít nhất, dòng nào dùng reference thì phải đánh số của reference vô trước dòng đó

1. Courses.lumenlearning.com. (2019). *What is a Feasibility Study/Report | Technical Writing*. [online] Available at: https://courses.lumenlearning.com/alamo-technicalandbusinesswriting/chapter/unit-4-b\_feasibility-report\_readings-2/ [Accessed 20 Aug. 2019].
2. Smartsheet. (2019). *What's the Difference? Agile vs Scrum vs Waterfall vs Kanban*. [online] Available at: https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban [Accessed 20 Aug. 2019].
3. softwaretestinghelp. (2019). *what-is-stlc-v-model*. [online] Available at: https://www.softwaretestinghelp.com/what-is-stlc-v-model/ [Accessed 20 Aug. 2019].
4. Software Testing Books. (2019). *What is Spiral Model? Advantages and Disadvantages of Spiral Model - Software Testing Books*. [online] Available at: http://softwaretestingbooks.com/what-is-spiral-model [Accessed 20 Aug. 2019].
5. Tutorialspoint.com. (2019). *SDLC - RAD Model*. [online] Available at: https://www.tutorialspoint.com/sdlc/sdlc\_rad\_model.htm [Accessed 20 Aug. 2019].
6. model, S. (2019). *Spiral model*. [online] IONOS Startupguide. Available at: https://www.ionos.com/startupguide/productivity/spiral-model/ [Accessed 20 Aug. 2019].
7. Simplilearn.com. (2019). *Why a Feasibility Study is Important in Project Management*. [online] Available at: https://www.simplilearn.com/feasibility-study-article [Accessed 20 Aug. 2019].
8. Startrungrow.com. (2019). *Purpose of Feasibility | StartRunGrow - your one-stop site for everything business*. [online] Available at: http://www.startrungrow.com/information/business/1,2315,purpose-of-feasibility.htm [Accessed 20 Aug. 2019].
9. Khan, M. (2019). *A Comparative Review of Shared Hosting vs Dedicated Hosting vs Cloud Hosting (2019)*. [online] The Official Cloudways Blog. Available at: https://www.cloudways.com/blog/wordpress-shared-vs-dedicated-vs-cloud-hosting/ [Accessed 20 Aug. 2019].
10. Automation.com. (2019). *Understanding Microsoft's .NET Technology & Its Impact on Automation Application Development*. [online] Available at: https://www.automation.com/library/articles-white-papers/manufacturing-intelligence-industrial-information-management/understanding-microsofts-.net-technology-its-impact-on-automation-application-development [Accessed 29 Aug. 2019].
11. Medium. (2019). *Java vs .NET: Factors to Consider*. [online] Available at: https://codeburst.io/java-vs-net-factors-to-consider-cde1d22b06a7 [Accessed 29 Aug. 2019].
12. Slant, E., Code, V., IDEA, I., Rider, J. and IDE, N. (2019). *Slant - Eclipse vs Visual Studio detailed comparison as of 2019*. [online] Slant. Available at: https://www.slant.co/versus/1957/12045/~eclipse\_vs\_visual-studio [Accessed 29 Aug. 2019].
13. GeeksforGeeks. (2019). *Difference between Waterfall Model and Spiral Model - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/difference-between-waterfall-model-and-spiral-model/ [Accessed 29 Aug. 2019].