DUY TÂN UNIVERSITY
International School

Capstone Project 2

CMU-SE 451

Architecture Design

Version 1.2

Date: 08 May 2021

SENIOR PROJECT MANAGEMENT SYSTEM FOR INTERNATIONAL SCHOOL

Submitted by

Tien, Nguyen Van

Phuoc, Ha Duc

Huy, Truong Dong

Dat, Nguyen Thanh

Approved by

Chau, Truong Ngoc

Proposal Review Panel Representative:

Name Signature Date

Capstone Project 2- Mentor:

Name Signature Date

PROJECT INFORMATION

Project acronym	Senior Project Management System for International School		
Project Title	SPMS		
Start Date	18 Feb 2022	**End Date**	15 May 2022
Lead Institution	International School, Duy Tan University		
Project Mentor	Chau, Truong Ngoc		
Scrum master / Project Leader & contact details	Tien, Nguyen Van Email: cnnguyenvantien@gmail.com Tel: 0704.042.832 Student ID: 24211208536		
Partner Organization			
Project Web URL			
Team members	**Name**	**Email**	**Tel**
24211202634	Phuoc, Ha Duc	dphuoc432000@gmail.com	0961622464
24211206538	Huy, Truong Dong	huydongtruong@gmail.com	0358040650
24211206470	Dat, Nguyen Thanh	ngthanhdat521@gmail.com	0767836541

REVISION HISTORY

Version	Date	Comments	Author	Approval
v1.0	26/02/2022	Initial Release	Dat	x
v1.1	05/05/2022	Update document	Dat	x
v1.2	08/05/2022	Format document	Phuoc	x

Table of Contents

Table of Figure

Table of Tables

SIGNATURE

Document Approvals: *The following signatures are required for approval of this document.*

Chau, Truong Ngoc *Mentor*		Date:
Tien, Nguyen Van *Scrum Master*		Date:
Phuoc, Ha Duc *Product Owner*		Date:
Huy, Truong Dong *Member*		Date:
Dat, Nguyen Thanh *Member*		Date:

1. Introduction

1.1. Project overview

The SPMS system is a system that could help the lecturer manage students' capstones, manage student workflow, grades and communicate plans to students in a timely manner. Fast and accurate way to save time and effort. Students can track and understand their capstone process.

1.2. Purpose

This specification covers following:

- Brief specification of the project, high level requirement.
- Detail quality attribution.
- System context, sequence diagrams.
- Architecture presented by various view types: Component and Connect, Module view and Allocation view.

2. Architecture driver

Business Problems:

- Students need assistance with details of a capstone.
- Users need an automated system to support submission, comment, assessment, project information or score information.

Business Need:

- A students can communication with group or mentor.
- Support for their students can assign, see project & document template when they need it.
- Students can view group details, score, evaluate and notification.
- Administrators can manage account user, project template and document.
- Teacher can manage students, mentor, group, new & announcements, defenses, schedule, project.

2.1. Business constraints

- Sources: 4 people.
- Project was started on: 15/02/2022.
- Project will be ended on: 15/05/2022.
- Project will be finished in 90 days (1440 hours).
- Cost: $3680.

2.2. Technical constraints

- **Database:** Postgres SQL.
- **Back-end:**

- Programming Language: Javascripts.
- Framework: Express (NodeJS), Nodemon.
- Libraries: Node-Postgres.

- **Front-end:**

- Programming language: HTML, CSS, Javascript.
- Framework: React, Hook.

- **Client:**

- Operating System: Windows.
- Web Browser: Chrome.

2.3. Function requirements

References to Product Backlog specification of ProductBacklogV1.0.docx.

2.4. Quality attributes

2.4.1. Utility table

There are following quality attributes that drive the design of architecture. Each quality attribute scenario is ranked with importance (I) defined by the Product Owner, and the estimated level difficulty (D). Both values are based on a scale of High (H) - Medium (M) - Low (L).

2.4.2. Quality attributes

2.4.2.1. Security

Table 2.4.2.1: *Security.*

Scenario: When user login into the system, the system will ensure security with user's account.	
Type	Security
Stimulus	Ensure security in the account.
Source of stimulus	User
Environment	Normal
Artifact stimulated	System
Response	Encode password
Response measure	Account protection

2.4.2.2. Usability

Table 2.4.2.2: *Usability.*

Scenario: When user has logged in successfully, the system will save user's session.	
Type	Usability
Stimulus	Save user's session
Source of stimulus	User
Environment	Normal
Artifact stimulated	System
Response	User access into the system
Response measure	User don't need to login more

2.4.2.3. Correctness

Table 2.4.2.3: *Correctness.*

Scenario: When the user performs operations, the corresponding information will be updated correctly.	
Type	Correctness
Stimulus	Performs operations
Source of stimulus	User
Environment	Normal
Artifact stimulated	System
Response	Change the information
Response measure	The corresponding information will be updated correctly.

2.4.2.4. Performance

Table 2.4.2.4: *Performance.*

Scenario: When user access the system, the requests from user will be processed maximum 3 seconds.	
Type	Performance
Stimulus	The requests from user will be process quickly
Source of stimulus	User
Environment	Normal
Artifact stimulated	System
Response	Data will display
Response measure	Maximum 3 seconds

2.4.2.5. Modifiability

Table 2.4.2.5: *Modifiability.*

Scenario: When user access into the system, user want to operate easily with features.	
Type	Modifiability
Stimulus	Update the User Interface after modifying data.
Source of stimulus	User
Environment	Normal
Artifact stimulated	System
Response	Make modification without affecting other functionality, Test modification, Deploy modification
Response measure	Repair time not exceed 3 hours (Time interval when the system is in degraded mode)

2.4.2.6. Availability

Table 2.4.2.6: *Availability.*

Scenario: When user access into the system, the website can usually access.	
Type	Availability
Stimulus	User access usually the website
Source of stimulus	User
Environment	Normal
Artifact stimulated	System
Response	Usually access
Response measure	20 hours / day

3. Architecture overview

This section shows the diagrams which bounds our target system and describes the architecture and interaction between components

3.1. System context

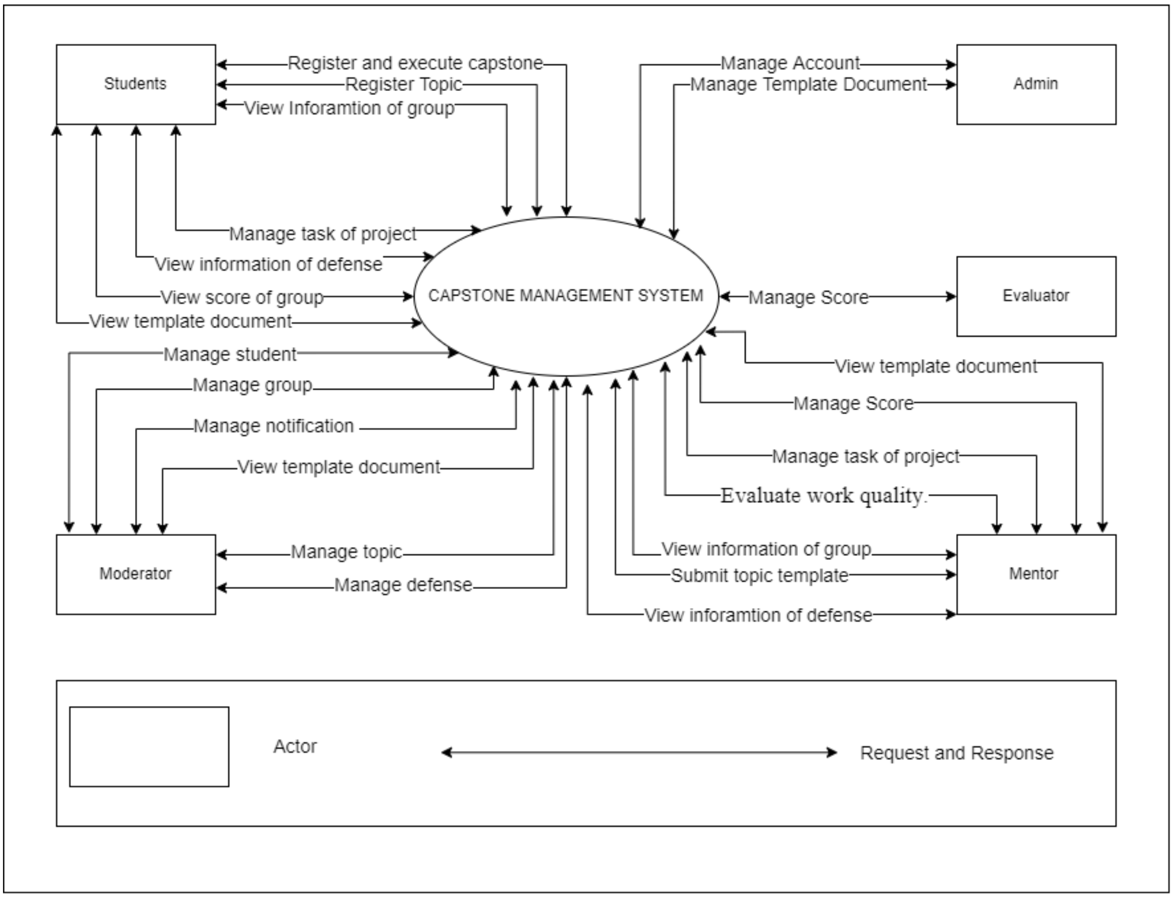

Figure 3.1: *Context Diagram.*

- Manage account:

- Admin can add, update, delete account of user.

- Manage template topic:

- Admin can upload, delete template document.

- Student can view and download template document.

- Register and execute capstone:

- Student can fill in form information and submit it to register execute capstone

- Student will wait to moderator approve and system will send account to mail of student.

- Register topic:
- Student can fill in form register topic and submit it to register topic for project.
- Student can choose topic template of mentor in topic template list.
- Moderator will approve and student can use it for capstone project.
- Manage task for project:
- Student can create stage for project.
- In each stage student can create task and assign it for members.
- In each task member can comment issue and report task done or late.
- Mentor can see stage and comment in each task of project.
- Manage student:
- Moderator can add, update, delete student.
- Moderator can approve for student can execute project and system will send account for student.
- Manage group:
- Moderator can create group and divide student and mentor for each group.
- Moderator can update, delete and export file excel group list.
- Manage defense:
- Moderator can divide defense and assign positions to each person.
- Moderator can update, delete and export file excel defense list.
- Manage topic:
- Moderator can view topic template list and topic of student list.
- Moderator can approve for topic of student.
- Manage notification:
- Moderator can create, update, delete notification.
- Submit topic template:
- Mentor can upload file topic template or fill in form topic template for student.
- Manage score:
- Mentor and Evaluator can input score of each member of group.
- Evaluator can export file summary score.

- Evaluate work quality:

- The system will aggregate the percentage of each member's contribution and sum it up.
- Mentor can see and evaluate work quality of each member.

3.2. Component and connector

- We mainly used a C&C view to argue and reason about architectural properties, quality attribute requirements, and functional requirements that the system must add here.

- This view type partitions the system into components that have some runtime presence such as processes, objects, data stores, and connectors or that represent pathways of communication such as data flows and access to shared storage.

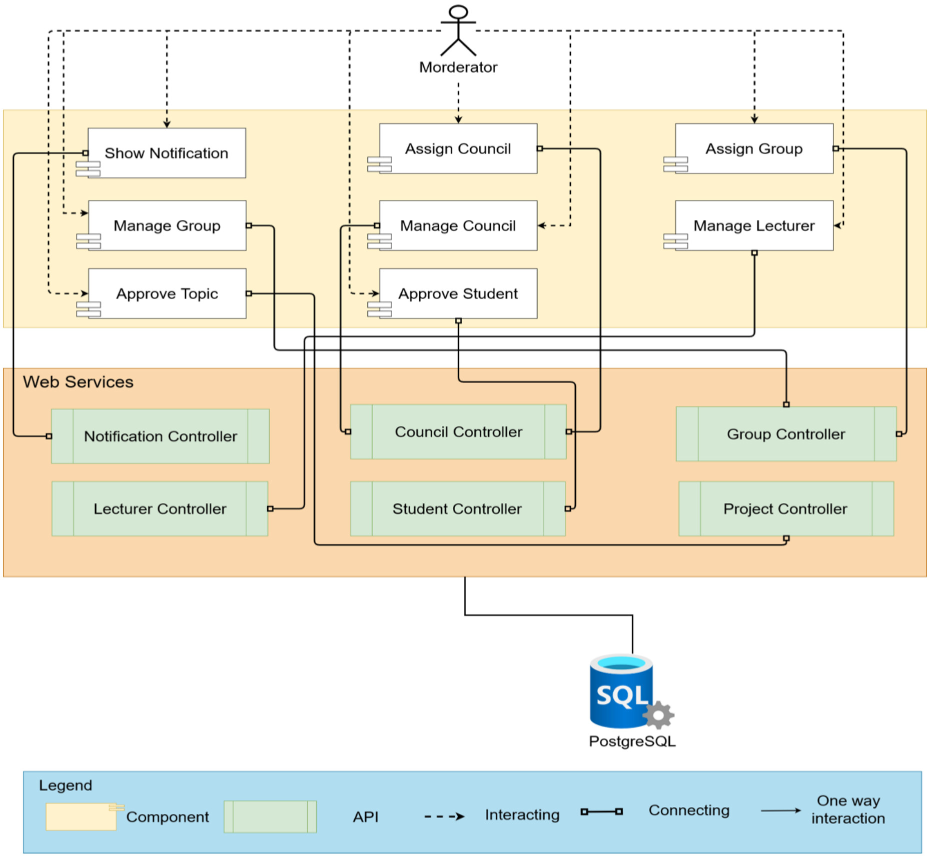

Figure 3.2.1: *Component & connector for moderator.*

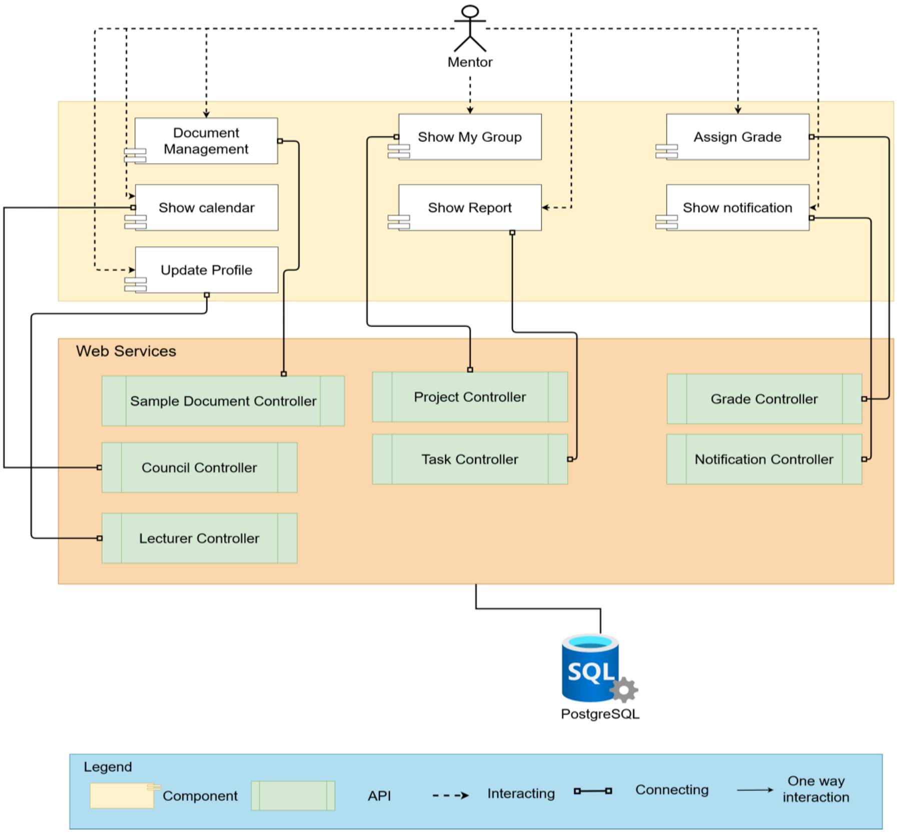

Figure 1.2.2: *Component & connector for mentor.*

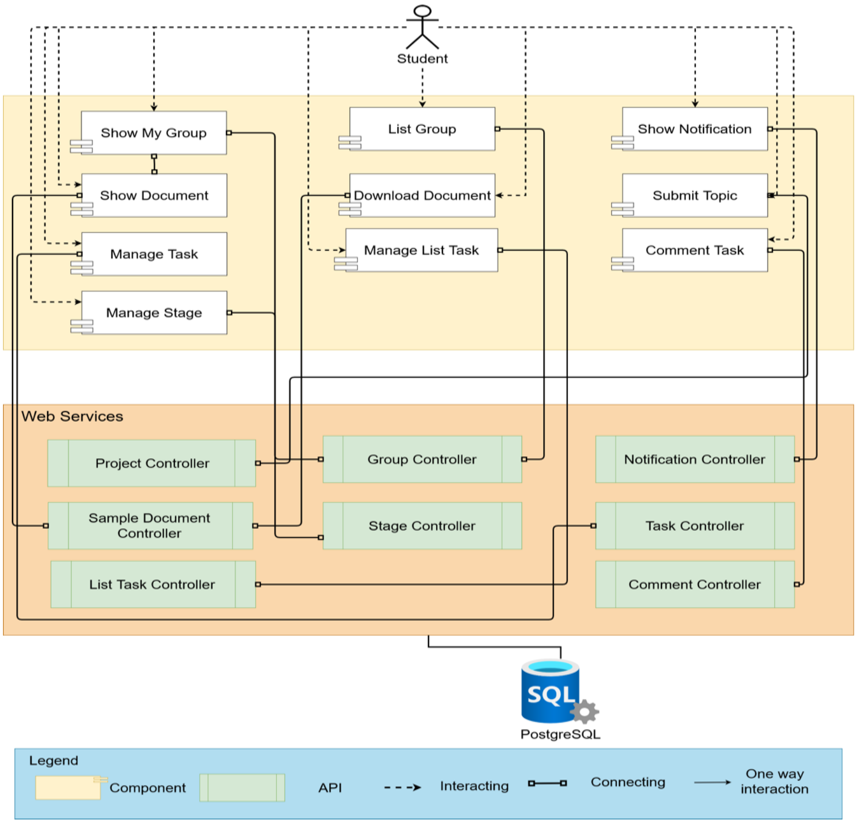

Figure 3.2.3: *Component & connector for student.*

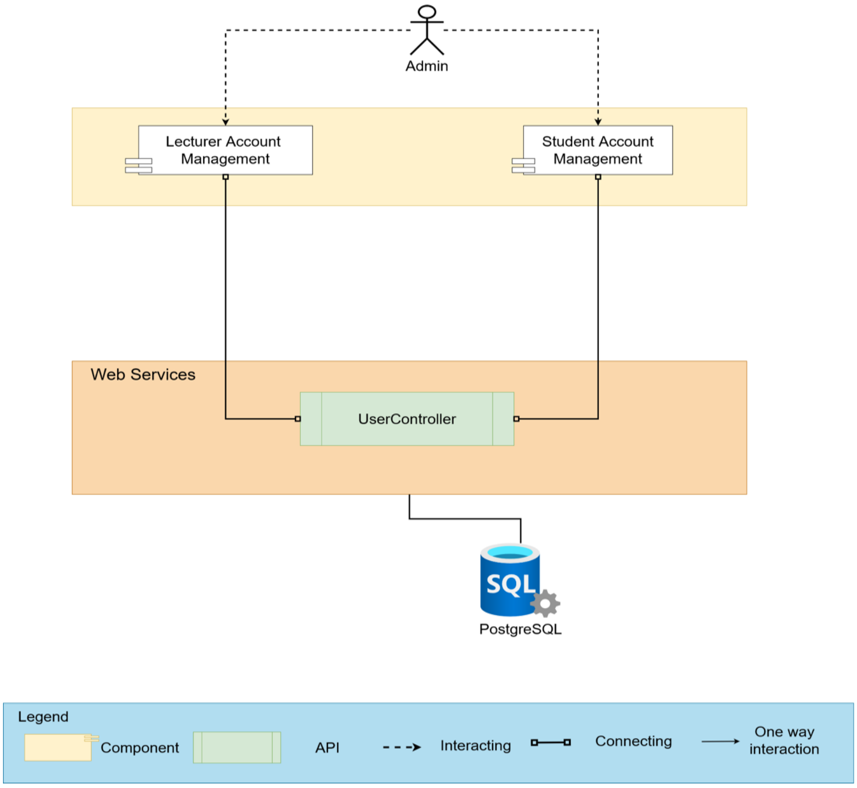

Figure 3.2.4: *Component & connector for admin.*

Prose

Table 3.2: *Component & connector prose.*

Element	Responsibilities
Show notification	User can see the notifications in the website.
Assign Council	Moderator can assign council into a group.
Assign Group	Moderator can assign group into a lecturer.
Manage Group	Moderator can add, update, remove group and show a list of groups.
Manage Council	Moderator can add, update, remove council and show a list of councils.
Manage Lecturer	Moderator can add, update, remove lecturer and show a list of lecturer.
Approve Topic	Moderator can approve topic from the student.
Approve Student	Moderator can approve student account from the student.
Document	Mentor can add, edit and remove files and folders.

Management	
Show My Group	Mentor and Student can see their group.
Assign Grade	Evaluator can assign grade for the students in a group.
Show Calendar	Mentor and Student can see the schedule of their councils.
Show Report	Mentor can see statical reports by every stages.
Update Profile	Lecturer can update their information.
List Group	Student can see all groups.
Show Document	Student can see sample documents.
Download Document	Student can download the sample documents.
Manage Task	Student can add, edit and remove tasks and assign tasks for members.
Manage List Task	Student can add, edit and remove a list of tasks.
Comment Task	Student can comment the tasks.
Manage Stage	Student can add, edit and remove a stage.
Lecturer Account Management	Admin can edit the lecturer accounts and reset password.
Student Account Management	Admin can edit the student accounts and reset password.

3.3. Module view

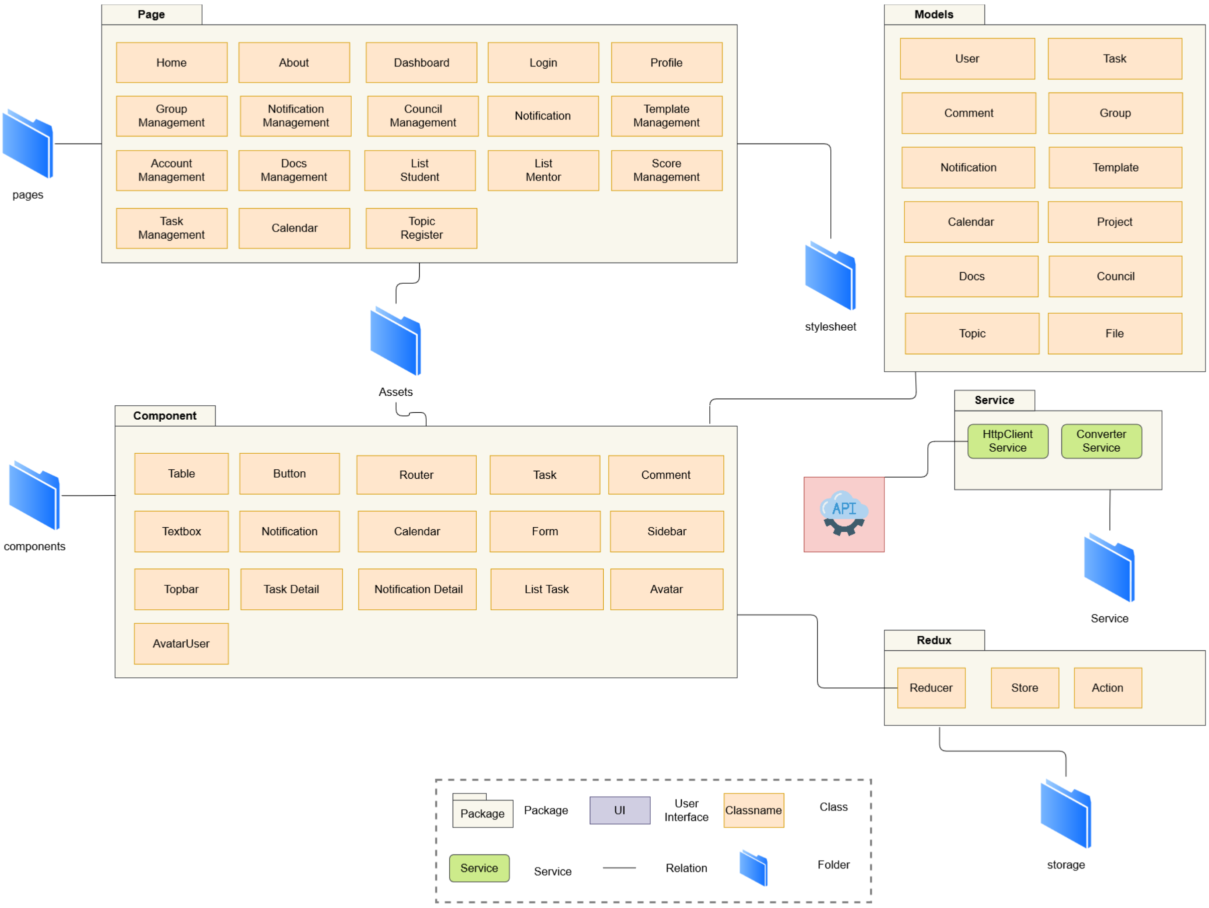

Figure 3.3.1: *Module view for client.*

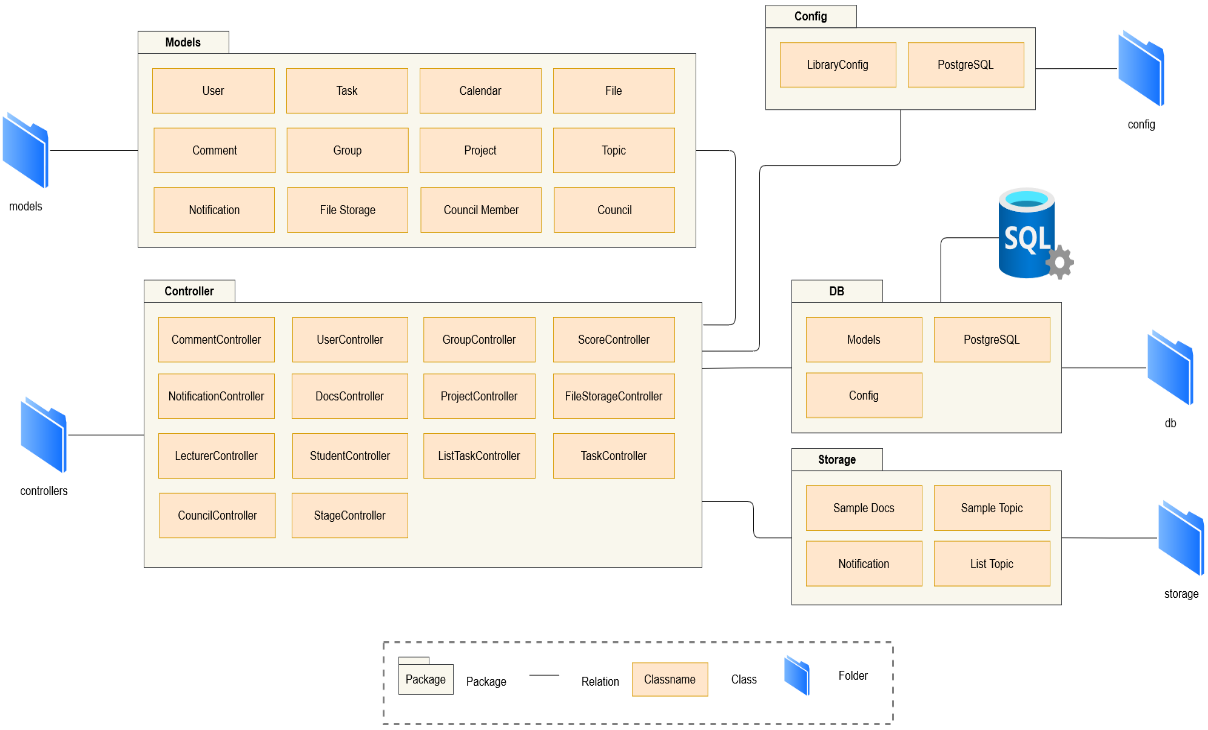

Figure 3.3.2: *Module view for server.*

Prose

Table 3.3: *Module view prose.*

Element	Responsibilities
Web View	The website package contains management modules for admin, students, mentor, moderator.
Models	Object data modules connect to tables in the database for transmission to the system interface.

Description

The system includes web view, services, and models. After users interact with UI. Data will be requested to the corresponding service, then will direct to model. Data be responded will be displayed on UI.

3.4. Allocation view

The allocation view models the run-time architecture of a system. It shows the configuration of the hardware elements when the system is deployed.

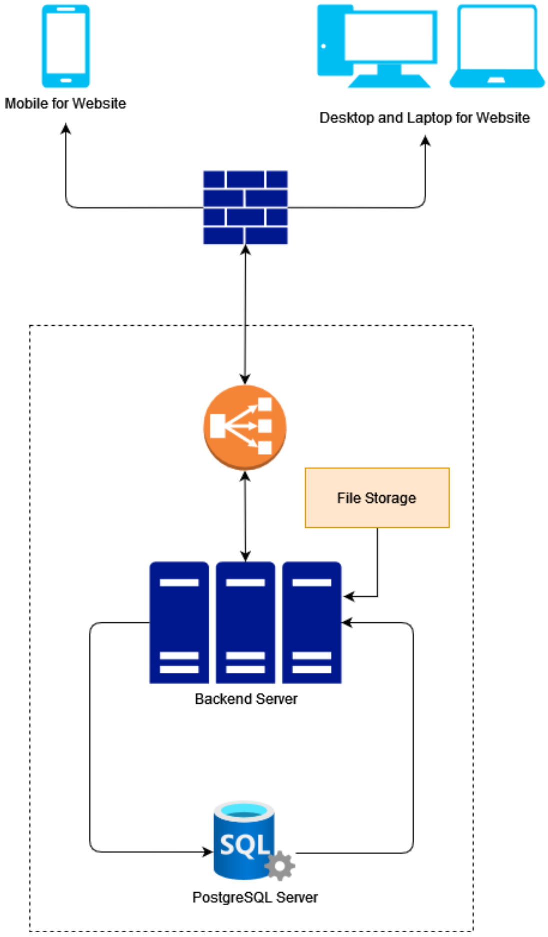

Figure 3.4: *Allocation view.*

Prose

Table 3.4: *Allocation view prose.*

Element	Responsibilities
Laptop or PC	Device running browser and helping Admin, Students, Mentor, Moderator to use the functions of the website to manage.
Backend Server	Provide an API to support the interaction between the user interface and the server. where to install and run the backend API.
PostgreSQL Database	The place contains all data about tours, user information... It is organized in tabular form.

Description

The system is deployed on web environment (using React JS library). They interact with the server through APIs to read and write data from the PostgreSQL database. In addition.

4. References

[1] Design standards, Document standards.

https://www.softwarearchitecturebook.com/svn/main/slides/ppt/26_Standards.ppt.

https://standards.ieee.org/standard/1471-2000.html.

https://ieeexplore.ieee.org/document/917550.

[2] Patterns.

https://en.wikipedia.org/wiki/Architectural_pattern.

[3] Evaluation standards.

https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:42030:ed-1:v1:en.

https://gabrielfs7.github.io/software-architecture/2019/10/18/atam-analyze-evaluate-architecture/.