SIM 3150 Web Application Development

ADMINISTRATOR DASHBOARD MODULE

GROUP MEMBERS:

HANIS SHAHFIQAH MUHAMAD HANAFI 179358
CHAN KAI FUNG 179960
MUHAMMAD ALIF BIN AZALI 180159
MUHAMMAD FAKHRULLAH BIN AHMAD 180670
ANIS NAJIB 177430

UNIVERSITI PUTRA MALAYSIA | SOFTWARE ENGINEER

PROJECT DESCRIPTION

This project basically to build as system which will help other company to generate directors report of the audit profit and loss account, and balance sheet. This system will require secretary to upload raw data from like ledger and profit and loss payment. This data will process by accountant and then will re-check by auditor. From all data will be process and the output will be the final report and can be view and print out by the company's director.

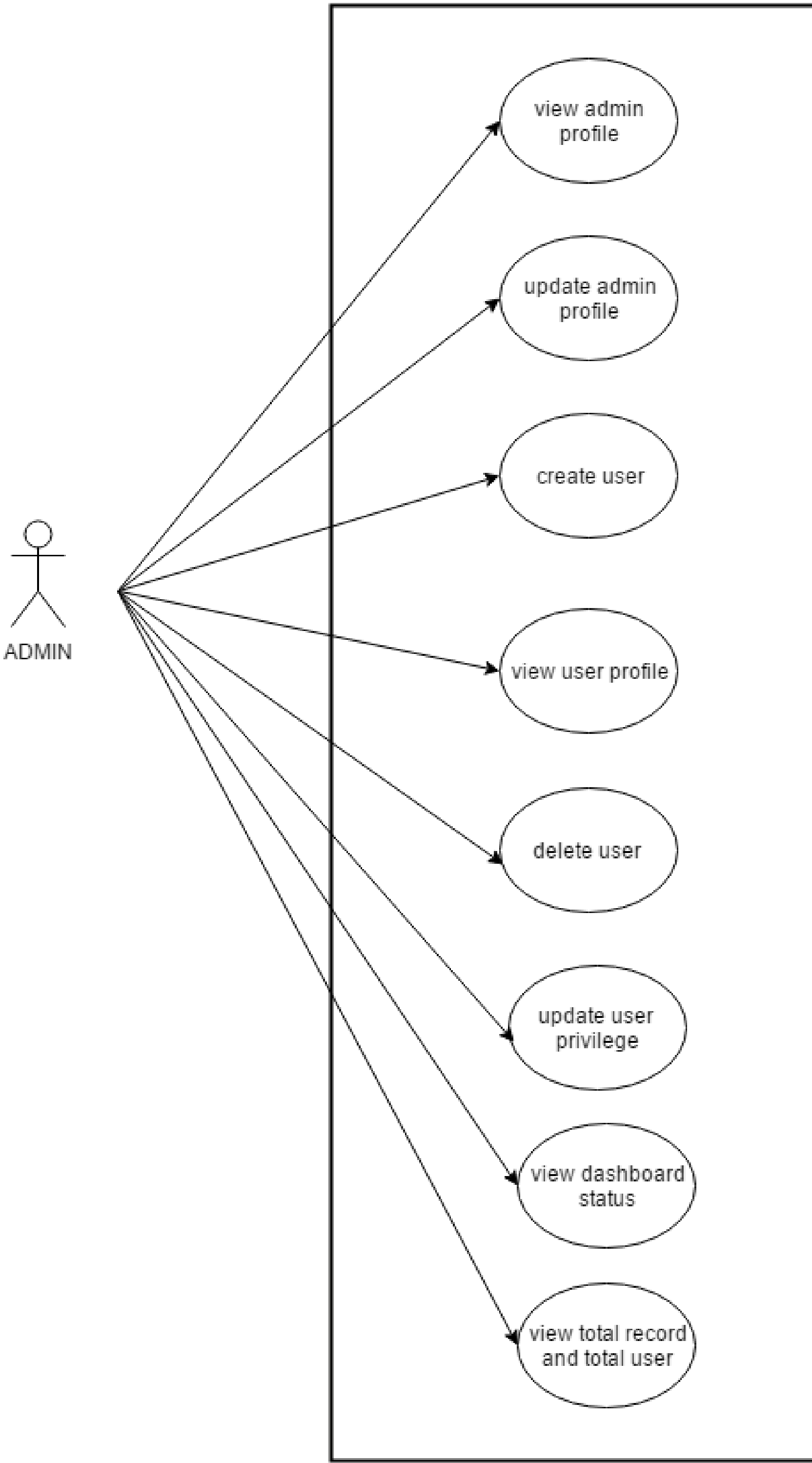
FUNCTIONAL REQUIREMENTS

- Admin shall able view his/her profile
- Admin shall able edit/update his/her profile
- Admin shall able create users which are accountants, auditors, directors, secretary, accountant`s delegate and auditor`s delegate.
- Admin shall able to view user's profile.
- Admin shall able delete user from the system.
- Admin shall able update user’s privileges.
- Admin shall able to view dashboard status.
- Admin shall able to view the total record entry and total number of user.

ADMINISTRATOR DASHBOARD MODULE

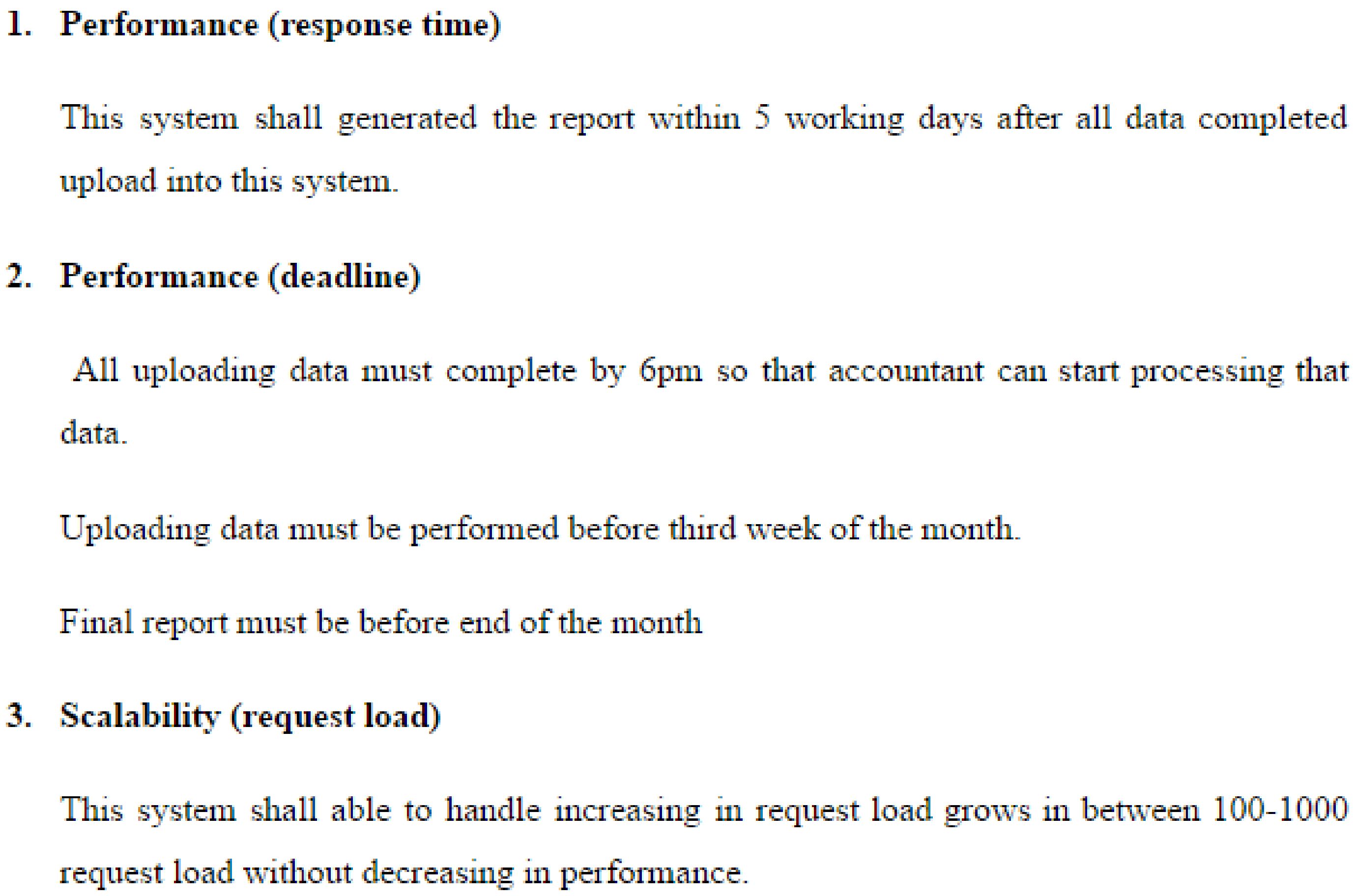
USE CASE

1.	View admin profile	Admin can view his/her own profile.
2.	Update admin profile	Admin can edit or update his/her profile.
3.	create user	Admin can add and create user which is accountant, auditor, accountant's delegate, auditor's delegate, company's director and company secretary.
4.	view user profile	Admin can view any user profile.
5.	delete user	Admin can delete user.
6.	update user privilege	Admin can give privilege to user based on their roles want.
7.	view dashboard status	Admin can view dashboard status
8.	view total record and total user	Admin can view the total of records entry and how many user that created.



ADMINISTRATOR DASHBOARD MODULE

NON-FUNCTIONAL REQUIREMENTS



1. **Performance (response time)**

   This system shall generated the report within 5 working days after all data completed upload into this system.

2. **Performance (deadline)**

   All uploading data must complete by 6pm so that accountant can start processing that data.

   Uploading data must be performed before third week of the month.

   Final report must be before end of the month

3. **Scalability (request load)**

   This system shall able to handle increasing in request load grows in between 100-1000 request load without decreasing in performance.

ARCHITECTURAL DESIGN - MODEL-VIEW-CONTROLLER (MVC) PATTERN

Laravel follows the model-view-controller (MVC) architectural pattern, which enforces a separation between “business logic” from the input and presentation logic associated with a graphical user interface (GUI). In the case of Laravel web applications, the business logic typically consists of data models for things like users, blog posts, and the GUI is just a web page in a web browser.

ARCHITECTURAL DESIGN - LAYERED ARCHITECTURAL STYLE

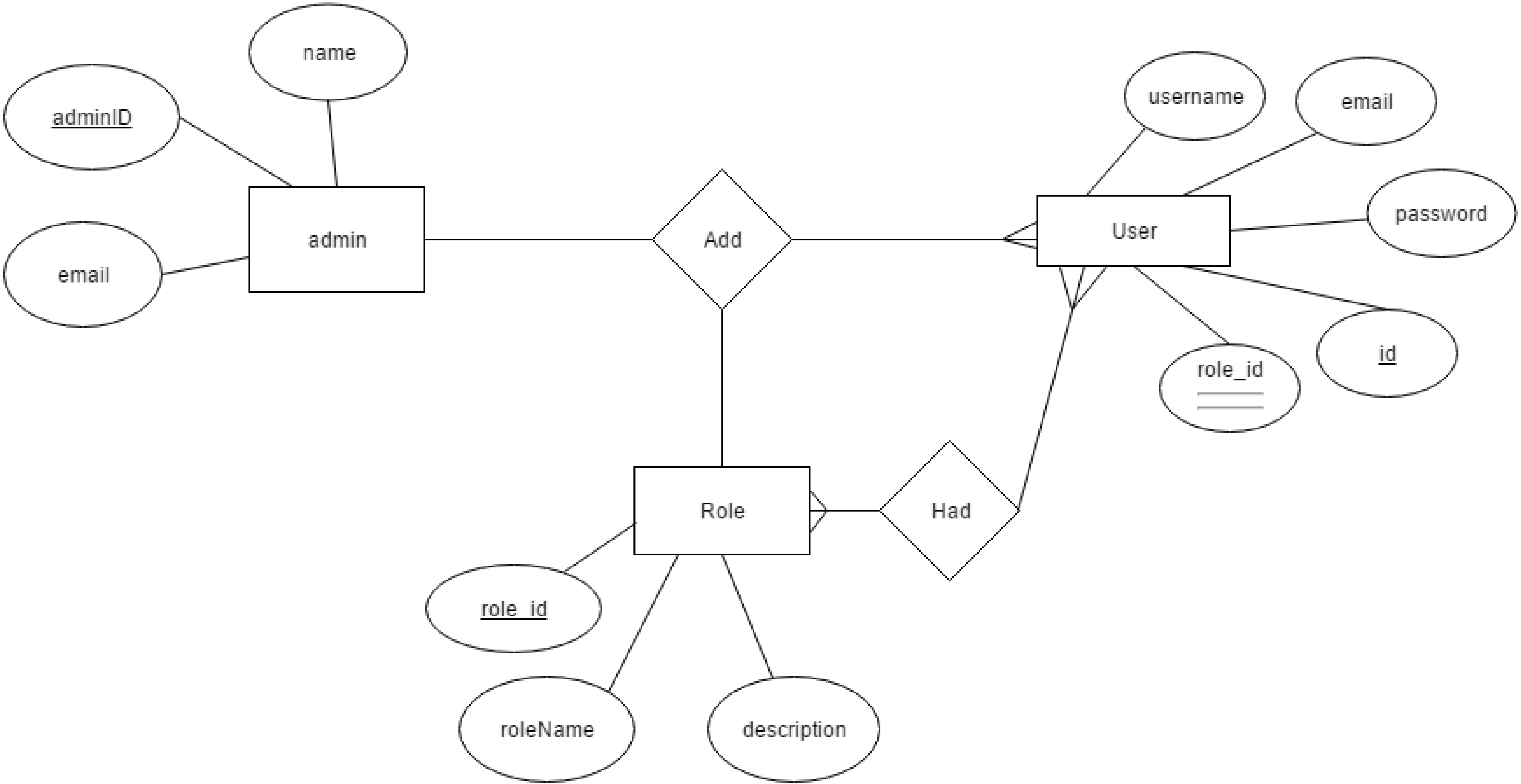
Layered architecture also used to developing this system. The first layer is user interface of the Administrator Dashboard. Second layer of this system is logic layer which coordinates the system, processes commands, makes logical decisions and evaluations and performs calculation. It also moves data between the two surrounding layers. The third layer is the storage layer. Data is stored and retrieved from the cloud database. The data is then passed back to the logic layer for processing and eventually back to the user. The code is arranged so the data enters the top layer and works its way down each layer until it reaches the bottom, which is database.

ARCHITECTURAL DESIGN - COMPONENT-BASED ARCHITECTURAL STYLE

For overall system, we implemented component-based architecture style where the system decomposition into six logical components. By using this type of style will help well-defined communication interface that contain methods, events and properties. This provides a higher level of abstraction than object-oriented design principle and does not focus on issues such as communication protocol and shared state.

ADMINISTRATOR DASHBOARD MODULE

ENTITY-RELATIONSHIP DIAGRAM



PUTRA MALAYSIA | SOFTWARE ENGINEER

HOW WE INTEGRATE WITH OTHER GROUP?

Since our module is admin dashboard, we need to obtain user's information from the user management and authentication module's database. Our module need user details such as user ID, name, email and role in order to determine user's privilege.

DEMO TIME!

THANK YOU!