

Catalogue

-----	Client meeting notes	-----
-----	Do, Be, Feel, Who table for the project	-----
-----	Cluster and content	-----
-----	Role Assignment	-----
-----	Sprint 1 Plan	-----
-----	Sprint 1&2 retrospective meeting summary	-----
-----	Sprint 1 review meeting summary	-----
-----	Sprint 2 Plan	-----
-----	Database model	-----
-----	Goal model	-----
-----	Product Backlog	-----
-----	Risk Management	-----
-----	Architecture Design - Logical view	-----
-----	Architecture Design - Component view	-----
-----	Architecture Design - Use case diagram	-----
-----	Architecture Design - Process View	-----
-----	Architecture Design - Physics View	-----
-----	Reuse Plan	-----
-----	Stand-up meeting notes	-----
-----	Project Links	-----

Client meeting notes

Client meeting notes - meeting 1 - week3

E-Portfolio Systems

- **Classical Musician Student - Melbourne Conservatorium of Music, The University of Melbourne**
- **We want the e-portfolio to showcase our student life as well as professional achievements. Some key**
- **information we would like to see in the e-portfolio include:**
 - **Name**
 - **Photos of me playing in the orchestra as a background**
 - **About me**
 - **Campus life - a section focuses on my involvement in the Melbourne Conservatorium of Music, for example: my performances at the University of Melbourne Symphony Orchestra, my charity events, UMSU activities**
 - **Music skills and achievements - for example ABRSM Piano Grade 8**
 - **Projects - videos and audios of my performances**
 - **Music sheets - some pieces and music that I composed or arranged. This section is only accessible to people who are members of my e-portfolio. Other website visitors can only see a sample.**
 - **Contact me - social media and email. It is preferable that my email is using a personal domain. Name to be determined.**

Client meeting notes - meeting 2 - week5

Website Name:

Hanan Davila

About Me:

Thank you for visiting my e-Portfolio! My name is Hanan Davila, and I am currently a final year student at the University of Melbourne, pursuing my Master's in orchestral performance. I am a professional pianist, but I am also learning guitar and flute. During my free time, I also enjoy composing music.

This e-Portfolio contains examples of my performance, music composition and other related experiences. Please feel free to explore my website.

Campus Life:

Last year, I successfully became an Orchestral Ensemble student at the Melbourne Conservatorium of Music, and participated in two out of four annual public performances, one of which was held overseas in Singapore. I'm also the secretary of the Music Student's Society at the University.

Music Skills and Achievements:

Piano: AMEB Grade 8, Professional level

Guitar: AMEB Grade 6

Flute: AMEB Grade 4

Projects (*Photos will followed*):

The University of Melbourne Symphony Orchestra 2018 - 2020

Fire Relief Charity Concert 2019

The Golden Stave Lunch Charity Concert May 2019

Profile Photo: Hanan Davila.jpg

It can be found on Slack - #client-requirements section

Background Orchestra Photo: orchestra.jpg

It can be found on Slack - #client-requirements section

Music Sheets:

Sheet_Music_1.pdf

Sheet_Music_2.pdf

Files can be found on Slack - #client-requirements section

Links to music (soundcloud):

<https://soundcloud.com/ashamaluevmusic/cinematic-piano>

<https://soundcloud.com/ashamaluevmusic/piano-cinematic>

For this one, please provide a player for visitor to immediately played in the websit

Client meeting notes - meeting 3 - week8

- **Overall design**
 - musical style & clean (change font and alignment)
- **Contact us section**
 - change to contact me
- **Connect with me section**
 - instagram & facebook
- **dashboard/project**
 - same theme with homepage
- **Project page**
 - 1. subheader
 - 2. cleaner style
 - 3. soundcloud aligned left
- **Nav header**
 - more visible
- **Upload page**
 - add background image
- **Homepage**
 - No animation on the top

Client meeting notes - meeting 4 - week9

1. Visitors should be able to contact without login.
2. Bug fix (some pages still have random white sides).
3. Add more music elements to where empty.(make the website look nicer)
4. Add youtube links upload function.

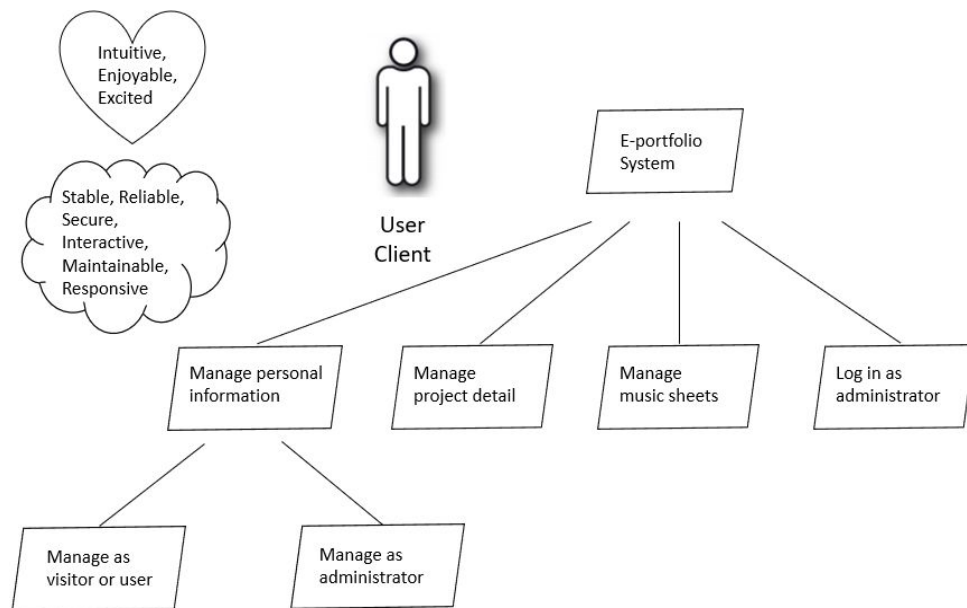
Do, Be, Feel, Who table

Do	Be	Feel	Who
Manage as visitor or user	Stable	Usable	Clients
Login as administrator	Scalable	attracted	scrum master
Manage personal information	Reliable	Interactive	Back end developer
Manage as administrator	Maintainable	Supported	Tester
Manage project detail	Efficient	Smooth	Maintainer
Manage music sheets	Responsible		

Cluster and content

Cluster	Content
Manage	Log-in and sign-up functionalities
	Manage music sheets
	Manage personal music links
	Manage personal information page
	Manage user information
	Manage music sheet page authentication
	Manage dashboard differentiation
	Manage designing attractive frontend
Learn	Set up database
	Image back-up in MongoDB
	API set up procedure
	File uploading to the database
	File rendering to the frontend
	GitHub pull request conflict resolution
	Authentication methods
	RESTful API usage
Understand	Contact client
	Basic requirements
	Client's tendency on page style
	Client's designing preferences
	Core elements of the application

Goal model



Role Assignment

Title	Responsibility	Who
scrum master	divide, supervise job	Ken Lin
testing leader	testing....	Siyu Ji
quality leader	coding standard...	Yannan Li
product owner	user story, user requirement	Xuelun Wang
achitecture leader	framework, plan, solutions	Yichao Yao

Sprint 1 Plan

Mission	Status
1. Implement the front end of homepage (including Name of the page, background photo, About me section, Campus life Section, Music skills and achievement section, Projects section, music sheet section and Contact me section.	Completed
2. Link sections to the main page of each section.	Completed
3. Implement a login and register system and its database.	Completed
4. Fill in some details of each section(client provided detail).	Completed
5. Connect MongoDB to node.js backend and create user model	Completed
6. Heroku cloud application set up	Completed

Sprint 1&2 retrospective meeting summary

2020.9.26 2:00pm AEST

Meeting called by: Yichao Yao

Facilitator: Zoom

Attendees: Yichao Yao, Xuelun Wang, Yannan Li, Siyu Ji, Ken Lin

Agenda item: Review Spring 1 goals and summarise sprint 2 goals

New Requirments
1. improve overall website style to be more cleaner and add more musical elements
2. Adjust contact us section
3. Adjust connect with me section: replace the github and codepen icon with their facebook link
4. Make the dashboard/project page look more formal
5. Improve front end of Project page <ul style="list-style-type: none">1. subheader (font/size)2. style (cleaner)3. soundcloud aligned left (alignment)
6. Can upload jpeg file to music sheet page

Sprint 1 review meeting summary

2020.9.23 1:15pm AEST

Meeting called by: Yichao Yao

Facilitator: Zoom

Attendees: Yichao Yao, Xuelun Wang, Yannan Li, Siyu Ji, Ken Lin

Agenda item: Review Spring 1 goals and summarise sprint 2 goals

Tasks	Degree of completion	Blockers
Client meetings were well organized	completed well	No
Goals reached - individual tasks accomplished	completed well	No
Tasks were finished on time	completed well	No
Documents were prepared with a high standard	completed roughly	Lack of knowledge about the documentation format
Clients were satisfied with our Sprint 1 outcomes	completed roughly	Front end design cannot fit client's taste
Sprint 2 tasks were discussed and well assigned	completed well	No

Sprint 2 Plan

Mission	Status
7. improve overall website design a. more musical style & cleaner (including font and alignment)	mostly done minor bug needed to be fixed
8. Adjust contact us section o change to contact me	done
9. Adjust connect with me section a. keep only instagram & facebook	done
10. Improve front end of dashboard/project page a. same theme with homepage (cleaner and grey/white)	waiting to merge new dashboard
11. Improve front end of Project page 1. subheader (font/size) 2. style (cleaner) 3. soundcloud aligned left (alignment)	done
12. Improve front end of Nav header a. visibility b. color	done can still improve by adding adaptive color
13. Upload page a. add background image	done
14. Improve front end of Homepage a. No animation on the top b. color (grey/white) c. font (size) d. alignment	done
15. Add a page for user feedback/advice - a. A user has to login to write these details b. Only clients can see these feedbacks via the	done

dashboard, and they can see a “feedback” section	
16. Implemente the SoundCloud link insertion backend	done

Sprint 3 Plan

Mission	Status
My message page fix view more.	done
fix the random white side on some pages.	done
visitors should be able to send message without login	done
Add more music elements to where empty.	done
Add youtube links upload function.	done

Database model

Account	
_id	String
first_name	String
family_name	String
email	String
password	String
is_admin	Boolean

Figure 1

Uploads	
_id	String
length	Int
chunkSize	Int
uploadDate	Date
filename	String
contentType	String

Figure 2

Message	
_id	String
first_name	String
family_name	String
email	String
password	String
phone number	Number
messages	String

Figure 3

Soundcloud_link	
_id	String
name	String
url	String
type	String

Figure 4

Youtube_link	
_id	String
name	String
url	String
type	String

Figure 5

Figure 1: model for user account information

Figure 2: model for files that are uploaded to the database

Figure 3: model for message from contact me

Figure 4: model for soundcloud links

Figure 5: model for youtube links

Product Backlog

Log in

As a	I want to be able to	So that	Priority
Visitor	Register an account	I can receive the latest news about the team via email and have access to view all the music sheets	Must-Have
Administrator	Get information about all registered users	Email them the latest events or music sheets of our team	Must-Have
Client	Register an account and having admin privilege	Different from normal visitors, I can have more options to personalize my page	Must-Have

Content management

As a	I want to be able to	So that	Priority
Administrator	Control the part of the content that visitors can see	Unregistered visitors can only view the sample music sheet	Must-Have
Administrator	update the projects, music sheets and latest information about our team	Present our latest work to visitors and registered users in time	Must-Have

Background

As a	I want to be able to	So that	Priority
Visitor and login user	Experience web interaction on the homepage	Get a simple and deep understanding of the team	Nice-To-Have

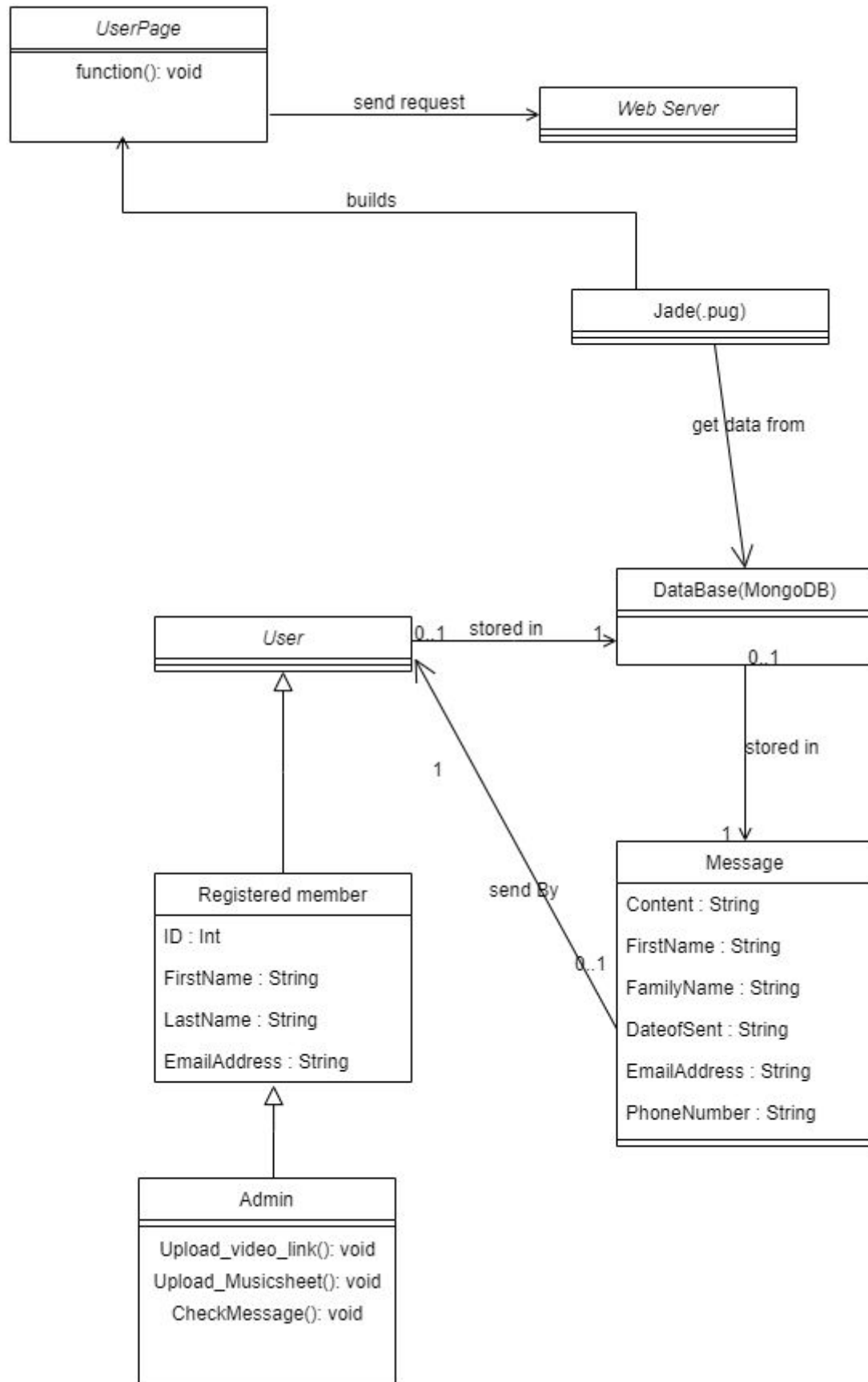
Risk Management

Risk Resource	Description	Suggestions
Risk Repository	Due to lack of experience working on git, team members will merge the work manually of one person without calling a pull request, or simply switch to master branch to push, this will possibly raise some vulnerability and dependency issues	Familiarize ourselves with pull requests and use pull requests if possible. Require confirmation of other team member with each merge commit
Unbalanced Workload	Lack of web developing experience and skills lead to an unbalanced workload of each team member, and this makes our front end and back end quite separated, affecting our working efficiency	Use trello more often and make wise decisions about workload allocation
Communication flaws	Using slack is not our first preference and we use weChat as our team communication tools, but our demonstrators and clients communicate with us via slack, and getting adapt to trello is also problematic for us as new developers, this affects our communication efficiency	regulate communication platforms
Expert Judgement	Our project is quite different from what others generally do, because of special client requirements. It may or may not raise bias/judgement from the lecturers/demonstrators since we're doing a totally different topic from the expectations	Keep connected with demonstrators and follow up client requirements
Lack of backend developers	Lack of experience, acquiring new technology stacks is time consuming, thus 1 person taking responsibility for all back end tasks is an issue for project diversity and problem solution limitations.	Encourage more team members to learn about backend skills
Cross-platform project operation	Some of the team members do not use github to upload files, and some do not use webstorm to work on our	regulate working processes and platforms

	current architecture. Most of the time our work is not synchronized, which means we're not on the same pace	
--	--	--

Architecture Design

Logical View



Development View

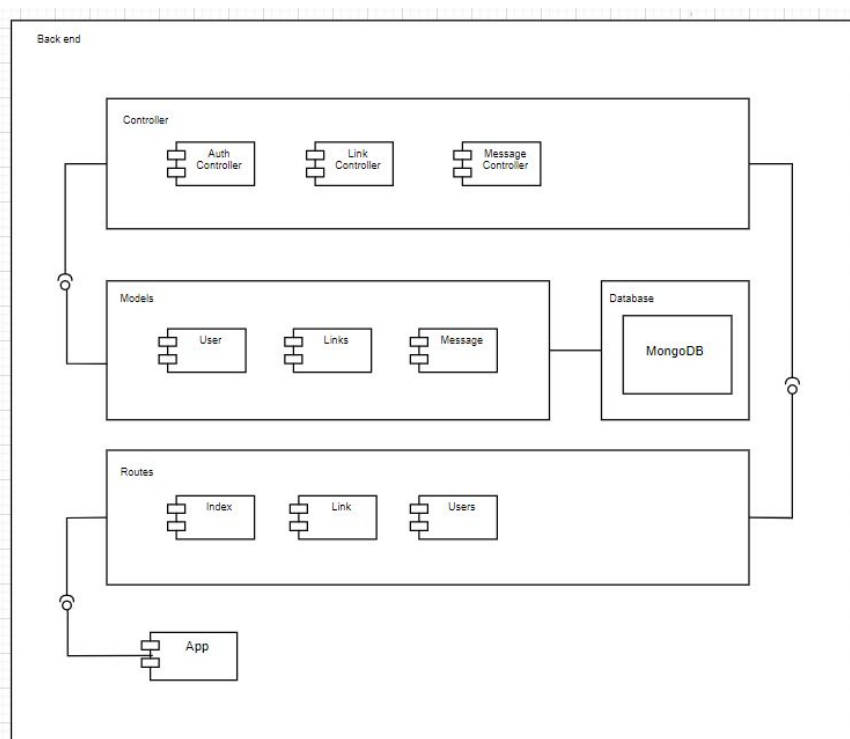
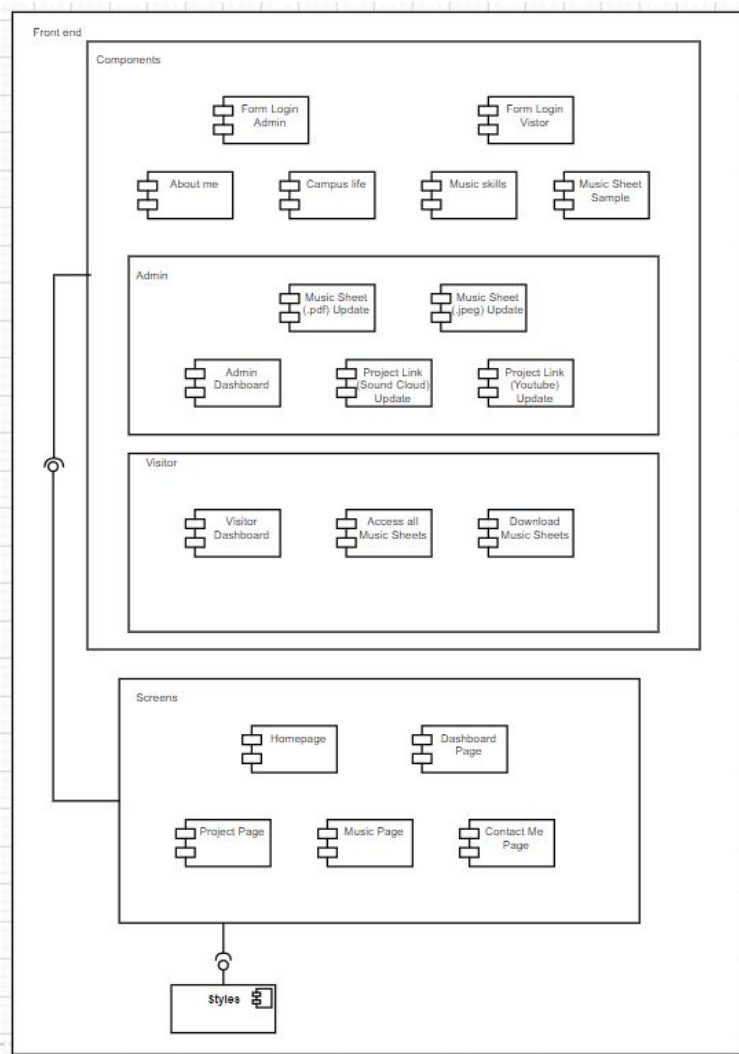
Component diagram: UML Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that are often used to model the static implementation view of a system.

Front-end introduction: For the front-end architecture, we have 3 main folders, the components, screens and style.

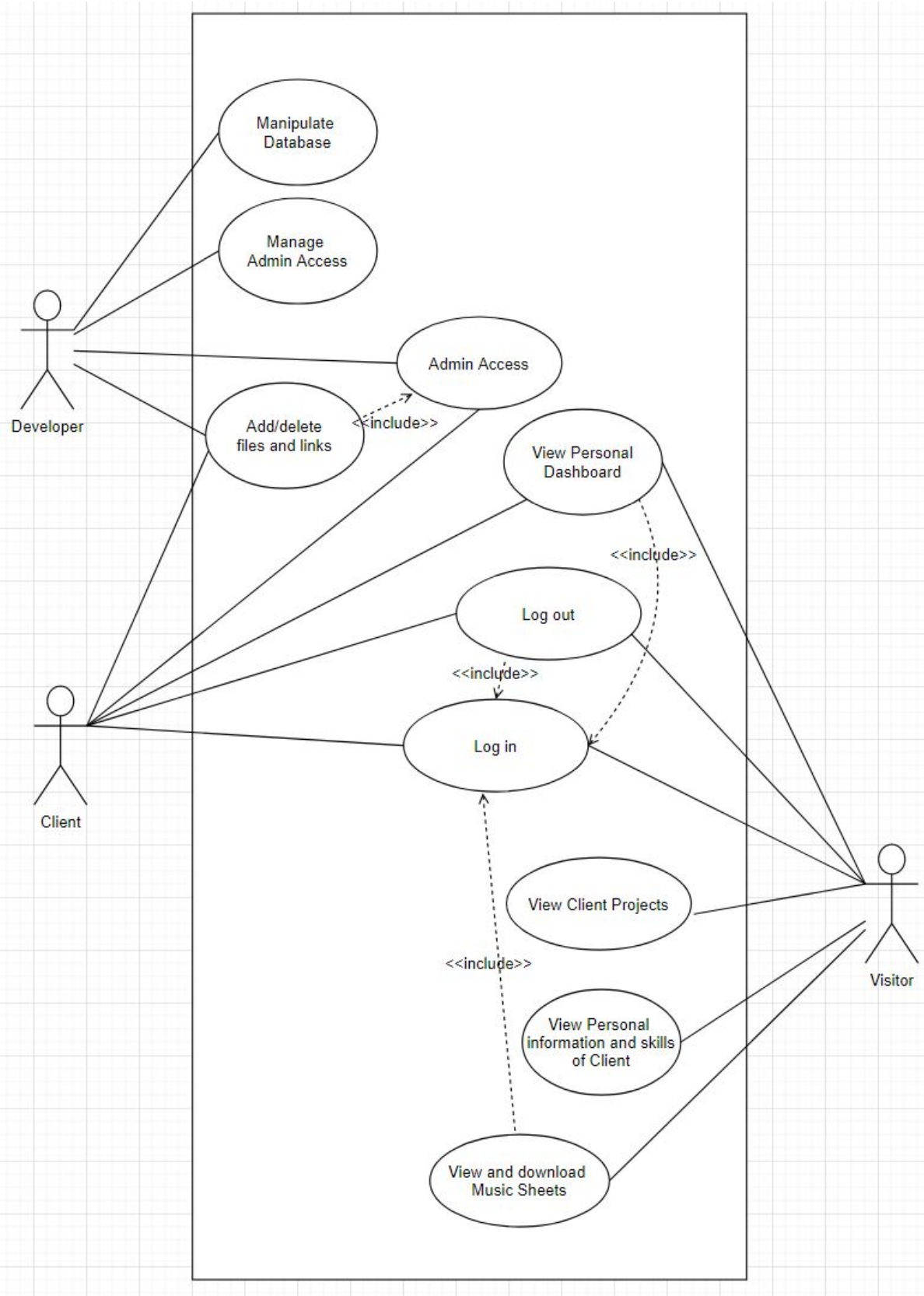
- **screens:** Contains the Pug and JavaScript code for each subpage. The duty of screens folder is to make sure every subpage can be displayed properly.
- **components:** Contains the reused components for every screen. In the Admin folder, we have components which are specifically for the administrators' operations instead of normal operations for visitors.
- **styles:** Including all CSS files for every component and screens. The screen or component can import the style file from this folder.

Back-end introduction: For the back-end architecture, we have 3 main folders, the controller, models and routes.

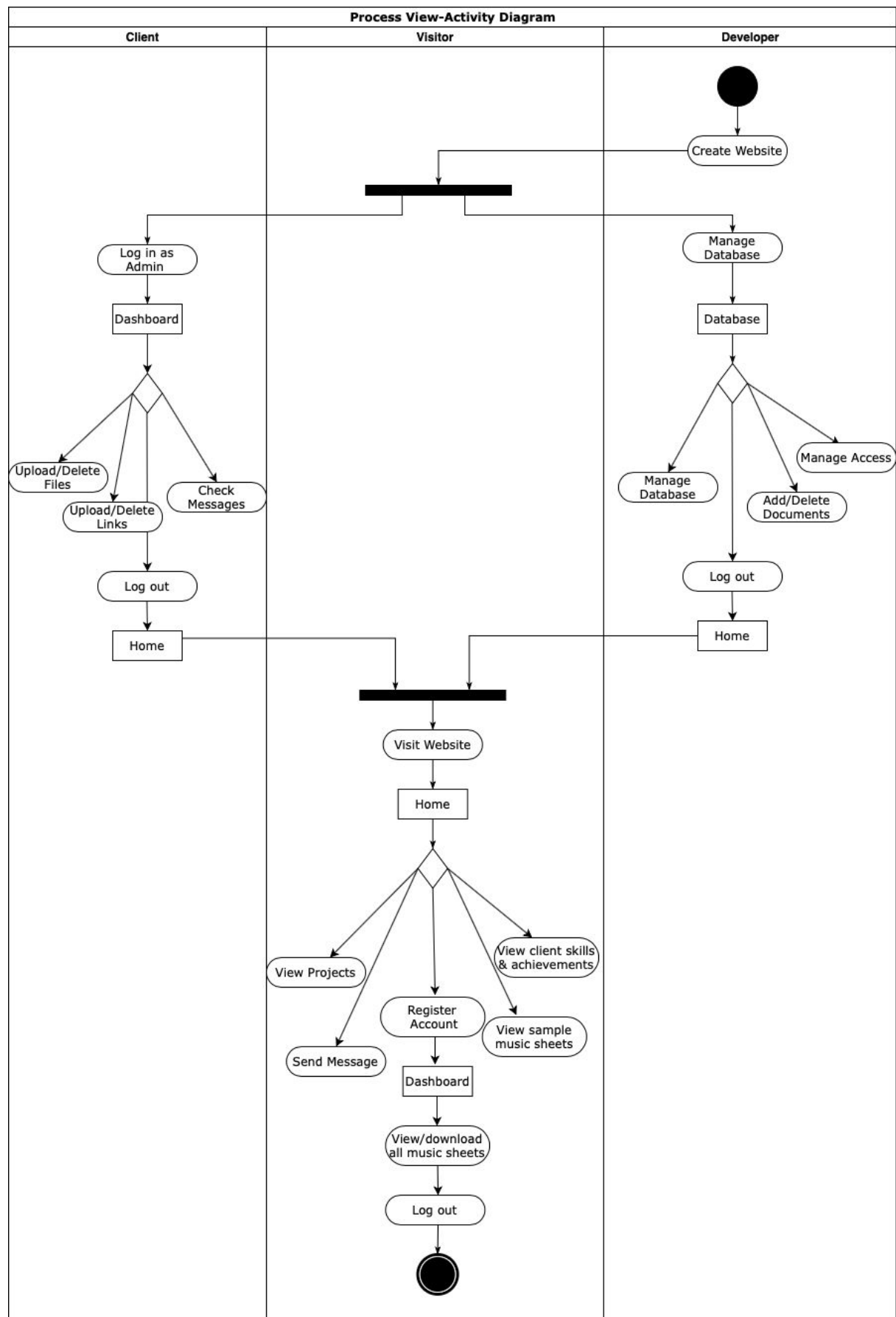
- **controller:** Contains all the functionalities of our back-end server that are routed by the front-end web page. There are 3 controllers in our application.
 - **auth controller:** It is used for defining the functionality of API calls related to authentication.
 - **link controller:** It is used for defining the functionality of API calls related to link update.
 - **message controller:** It is used for defining the functionality of API calls related to contact message.
- **models:** Contains the JavaScript files used for creating MongoDB database schema and for establishing the connection between server and database.
- **routes:** Contains the files that allow the API requests from the front-end to be routed to the specific controllers.



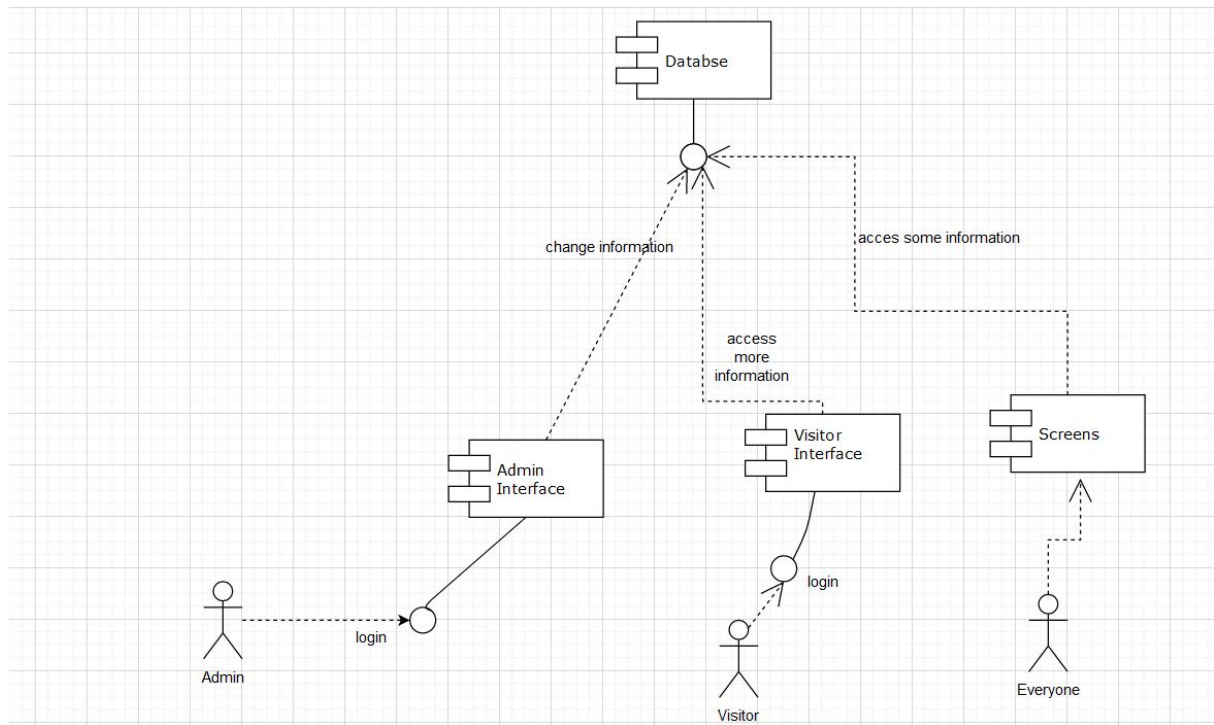
Use Case Diagram



Process View



Physics view



Reuse Plan

Component	Description	Notes	Documentation
Jade	Front-end framework	<ol style="list-style-type: none">1. Less complex syntax than HTML2. It improves readability, which means maintainability and productivity3. Jade has great support towards the Template inheritance. It means that it helps in managing templates that resemble object-oriented programming techniques.	https://jade-lang.com/
Node.js	Back-end framework	<ol style="list-style-type: none">1. Node.js offers an Easy Scalability2. Since JavaScript is one of the most popular programming languages, most of the front-end developers have a good grasp over it.3. Node.js has been regarded as a full-stack JavaScript for serving both the client and the server-side applications.	https://nodejs.org/en/
MongoDB	Databse	<ol style="list-style-type: none">1. MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.2. No complex joins3. Conversion/mapping of application objects to database objects not needed.	https://docs.mongodb.com/
Mongoose	elegant MongoDB object modelling node.js	We can interact with Mongo directly using Mongo Driver, Mongoose will simplify that interaction by allowing you to model relationships between data and validate them easily	https://mongoosejs.com/docs/api.html
RESTful JSON	HTTP APIs	<ol style="list-style-type: none">1. The REST protocol totally separates the user interface from the server and the data storage. This has some advantages when making developments.2. The REST API is always	https://restfulapi.net/

		independent of the type of platform or languages	
Heroku	Cloud Application platform	1. User-friendly to new developers 2. Enhance the productivity of cloud app development team	https://devcenter.heroku.com/
passport	Authentication System	Extremely flexible and modular, It can be unobtrusively dropped into any Express-based web application.	http://www.passportjs.org/
bcrypt	Encryption System	Bcrypt is able to mitigate different kinds of attacks by combining the expensive key setup phase of Blowfish with a variable number of iterations to increase the workload and duration of hash calculations.	https://www.npmjs.com/package/bcrypt

Project links

<https://hanandavila.herokuapp.com/>

<https://github.com/potatofrank/project-ePortfolio>