SOFTWARE ARCHITECTURE

IMT - eLearning

TEAM MEMBER

* 1351047 Huỳnh Minh Tú
* 1451012 Nguyễn Phú Cường
* 1451042 Pham Thi Bich Ngoc

The development environment requires the following installation.

Windows 10 recommended OS.

|  |  |
| --- | --- |
| **No** | **Language** |
| 1 | HTML |
| 2 | CSS |
| 3 | Typescript (Javascript) |
| 4 | SQL |

|  |  |
| --- | --- |
| **No** | **Framework** |
| 1 | Angular 4 (Front – end) |
| 2 | Bootstrap (Front – end) |
| 3 | Sails JS (Back – end) |

|  |  |
| --- | --- |
| **No** | **Database** |
| 1 | PostgreSQL |

|  |  |
| --- | --- |
| **No** | **OPEN SOURCE** |
| 1 | Socket.io 2.0 |
| 2 | Jitsi |

|  |  |
| --- | --- |
| **No** | **IDE** |
| 1 | WebStorm (recommended) |
| 2 | Visual Studio Code |

1. We apply **VSTS** to follow Agile Scum and manage tasks . 1 meeting per week.
2. We use **Git** is the most commonly used version control system which is attached in **VSTS** is also a tool.
3. We communicate through **Ryver** as the main chat tool.

**Server Side**

Why do I choose **Node Js** for back-end**?**

1. **The numbers**

* We can quite clearly see that Node.js is leading the race for number of modules.

1. **Non-opinionated**

* Node.js assumes nothing and gives you the bare minimum from a fresh

1. **Speed of development**

* The way Node works is that it makes you string together different components to build an application, this gives you flexibility but it takes considerably more time considering you need to download a module for practically everything but it can be acceptable.

1. **The Learning Curve**

* The beautiful thing about Node is that it is accessible to not only developers from backgrounds, it is also easy for front-end developers to pick up.
* Easy to learn

1. **Finding Talent**

* The major benefit of finding Node developers is that Node.js has reached a point where most front-end developers as well as back-end developers have used it in some form. Being able to hire a front-end developer who also knows Node means you're getting extra value for money and if you're a lean startup, money definitely matters.

Why do I choose **ExpressJS**?

Deciding on the Node.js web application framework for your project, you should pay attention to the following criteria:

* Community
* Documentation
* Versions upgrading
* Scalability
* System resources loading
* Performance
* Simplicity of development and testing
* Availability and variety of modules
* Customization

Firstly, Express.js has the biggest community.

Secondly, as for official documentation, Express.js docs are complete enough to help developers start off and provide support in tough cases.

Moreover, as Express does not limit the choice of additional tools, we can freely handle with the integration of push notifications, payment gateways, in-app purchases, social media, analytics and everything our clients may require.

#### ****Do learn Express before migrating to Sails****

## What is PostgreSQL?

**PostgreSQL is an open-source RDBMS**

Most modern applications require a flexible, scalable system like MongoDB. However, there are situations in which an RDBMS like PostgreSQL would be more appropriate. For example, applications that require complex, multi-row transactions (e.g., a double-entry bookkeeping system) would be better suited for PostgreSQL.

**Client Side**

JavaScript popularity continues its rising in **2017**.

Why do i choose **Angular 4 JS** instead of **React Js**?

1. Angular is a much fuller featured framework than React
2. Support for Angular is much, much better, with a larger community

I think both of them are good. Although React has better performance than Angular due to React’s implementation of a virtual DOM, I still choose Angular 4 for this project because there are many existing projects of IMT working with Angular.

**OPEN SOURCE**

**Socket.io**

## FEATURING THE FASTEST AND MOST RELIABLE REAL-TIME ENGINE

Why do I choose **Socket.io** instead of **Websockets** framework?

* Socket.io lets us emit **arbitrary events** with arbitrary data (even binary blobs) to the server.
* Socket.io lets the **server and client both** emit arbitrary events and handles all the parsing and packing/unpacking
* Socket.io gives us **abstractions** like rooms which aren't really part of the websocket spec so you will have to roll your own if you want to use websockets.
* Websocket client will not work on browsers that don't support it (think IE8, IE9, Opera Mini) but a socket.io client server pair will since it has **fallbacks**

**JITSI**

## a set of open-source projects that allows you to easily build and deploy secure videoconferencing solutions