TDT4140 – Software Engineering
Group 26: Preliminary Architecture

Sindre Hansen Vegard Helgesen Hesselberg
Eirik Rismyhr Stian Sørli

Spring 2017

Contents

1 Scalable web app

1.1 Description

Implementing the bot as a scalable web app

1.2 Stakeholder concerns

Concern from lecturers: Should work on both computers and smart phones.

1.3 Related user stories

User story # 9

1.4 Solution

Developing the bot as a scalable web app. Doing this, the bot will be easily accessed and utilized during lectures, as it's accessible through both laptops and mobile phones.

1.5 Considered alternative solutions

- Developing the bot as a native app running on either a computer or a mobile device.
- Developing the bot as a plugin to another existing application.

1.6 Influencing forces

Positive

- Accessibility for the end users
- Ability to run on most devices without requiring extra development time

Negative

- Variable internet connection can cause difficulties
- Application may not be well supported by outdated software

1.7 Evaluation

We believe this is the best solution of all considered, compatibility is incredibly important for an application such as this, as well as not forcing students and teachers to download a native app to run on their phone/computer.

2 Database

2.1 Description

The chat logs will be saved in a MySQL database.

2.2 Stakeholder concerns

What will be saved, how to get meaningful answers from the data.

2.3 Related user stories

User stories #3, #4 and #5.

2.4 Solution

For automatic saving of chatlogs for lecturers, they will be sent and stored structurally on a MySQL database server, where they'll be available for use

2.5 Considered alternative solutions

- Data could otherwise be saved locally on the end users' device
- A remote non-database server to save the chatlogs

2.6 Influencing forces

Positive

- Structured saving of valuable and useful data, letting the end users access and use it
- Easy to implement, as well as use for developers and end users

Negative

- Possibility of database going down, causing issues for end users.
- Internet connection required to fetch data.

2.7 Evaluation

A well known and popular platform for handling data. We think this is a good choice.

3 Web framework

3.1 Description

Django is a web framework based on python designed for quick creation and less code. It emphasises reusability, rapid development and the principle of not repeating yourself.

3.2 Stakeholder concerns

The teams' experience with the framework.

3.3 Related user stories

There are really no user stories that specifically relate to this. All of them has something to do with this choice.

3.4 Solution

Develop the app using Django to make use of the various framework features and make development quicker and easier.

3.5 Considered alternative solutions

Writing the web page using plain HTML, a PHP framework or a Javascript framework (eg. AngularJS).

3.6 Influencing forces

Positive

- Uses a lot of the developer's previous knowledge and skills
- Effective framework for developing the app we have in mind

Negative

- Developers have to learn and understand the framework prior to implementing

3.7 Evaluation

We are all familiar with python, so choosing a python framework seems like the obvious choice. Though we have no previous experience with the framework, we believe this framework will be very useful for developing our application, and understanding the framework will not take up too much precious development time.

4 Real-time messaging server

4.1 Description

We will use *Centrifugo* as our real-time messaging server.

4.2 Stakeholder concerns

Students chatting must be connected together, over a safe and fast connection.

4.3 Related user stories

Same as Django.

4.4 Solution

Centrifugo allows us to keep a connection open between the client and the server. It is also event based which is an essential feature for us.

4.5 Considered alternative solutions

- A polling-based framework
- Writing a server from scratch
- Twisted
- Tornado

4.6 Influencing forces

Positive

- Event-based solution, making it suited for a chat program

Negative

- Developers have to learn and understand the framework prior to implementing

4.7 Evaluation

It is a necessity for the product to be event-based, so to implement the web server in any other way would be detrimental to the project.

5 Static web server

5.1 Description

Static web content will be hosted on an NginX server.

5.2 Stakeholder concerns

The content hosted on the server has to be available at all times.

5.3 Related user stories

A part of the overarching functionality, and not connected to any specific user story.

5.4 Solution

For all static content which will not change as the end users use the application, having an NginX server to be stored on is appropriate.

5.5 Considered alternative solutions

- Apache
- Cherokee
- Gunicorn

5.6 Influencing forces

Positive

- Lets the application be loaded for users even if chat servers are down.
- Moves workload away from the designated chat servers

Negative

- Might introduce extra costs in development time and/or money

5.7 Evaluation

We're confident having a designated server for static content will benefit the end users greatly, as well as relieve a lot of stress from the chat servers. For this purpose, NginX delivers what we need.