

Project 362s

Tutorspace

Group System Report



Team Members:

Justin Jenecke	215163052
Gaynor Jantjies	213069555
Shameel Kiyang	217050743
Sihle Jijana	216273919
Kevin Karelse	219000859
Themba Khanyile	217238173

Contents

Team Members:	0
Contents	1
Introduction	2
User Documentation	4
Dolor sit amet	4
System Documentation	4
Dolor sit amet	4
References	4
Dolor sit amet	4

Introduction

Tutorspace tutor classifieds site

Website where tutors can create posts to advertise their tutoring services, and students can find tutors and assistance while studying remotely

Programming Stack:

- Java
- Spring-Boot
- MySQL
- HTML
- CSS
- JavaScript

1st Version of Project: WordPress Website

Online platform for tutors to advertise their services, and for students to find these tutors and/or create posts looking for a tutor for a specific subject. Limited to tertiary institution students

Version 1.5 of Project: PHP Website

Stack change 25-July-21 - Project was changed from PHP web application to static website

2nd Version of Project: Web Service and RESTful Client

Project was changed from a web application to a static website, as the server-side code will be built as a separate application, instead of being integrated with the website.

Reason for this is that frontend and backend code can be segregated, as they will have two different teams working on them simultaneously.

2nd Version of Project: Web Service and RESTful Client

Project downscale 22-September-21 - Backend was changed from web service to single entity API, frontend website was changed to SPA

- Backend API and database integration was built by team lead Justin Jenecke
- Frontend to be developed by rest of group
- landing tab by Sihle Jijana and Gaynor Jantjies
- Create Tutor tab by Kevin Karelse
- About page by Themba
- Contact tab by Shameel Kiyang
- Integrations by Justin Jenecke

User Documentation

What is the system about:

This system is all about assisting students and tutors by providing them with a platform to increase student's learning. Furthermore, the system gives tutors a medium to advertise and market their services during lockdown and to provide students that are struggling to study remotely an opportunity to get assistance in a more personal and focussed setting than an online classroom.

Who are the users:

The target audience for this system is mostly, but not limited to university and college students who are required to study remotely, the system will also be used by tutors offering tutoring services, and lastly any student who lacks the means or resources to self-study or study remotely efficiently.

Specific instructions for the users:

Tutor

As a tutor, to post your services to the website do the following:

1. Click on 'Create Post' tab
2. Enter your name, tutoring subject, and cost rate per hour in the form
3. Click 'submit' to submit the form
4. Information will be successfully added to the landing table

Student

As a student, to contact the tutor relevant to you do the following:

-
1. Click on 'All Posts' tab
 2. Click on 'contact tutor' next to the tutors name
 3. Fill in necessary information to contact the tutor
 4. Click on 'Submit'

About tab

To identify the developers who played a role in developing this website you will click on the 'About' tab.

Contact tab

If you need to contact us, do the following:

1. Click on the 'Contact' tab
2. Fill in the form with your name and email address
3. Click on 'Submit'

System Documentation

Single Model Entity

The system initially contained three main entities, namely: Student, Tutor and Post, with various aggregate entities forming across them to indicate dependencies and relationships (eg: Tutor-Post).

This system now compounds all this abstracted information into one large entity known as Post or posts made by Tutors for better referencing.

Public API Methods

Simple Spring-Boot API for Tutorspace

API entry point:

/ts/api/v1/post

returns String as a soft guide to different uri endpoints

UML

- + **Class Post**
- int postId;
- int studentNo;
- String firstName
- String lastName,
- String email
- String institution
- String major
- String subject;
- + Getters() and Setters()
- + ToString()

Post entity endpoints:

Get all Posts

URI: */ts/api/v1/post/all*

Details: Returns List of all Posts in database

Create Post

URI: */ts/api/v1/post/create*

Details: Request body must contain a Post object

(int student no, String firstName, lastName, email, phone, institution, major, subject)

Object Id is generated by Database

Read single Post

URI: */ts/api/v1/post/read/{id}*

Details: Returns a Post object by given Id

Path parameter must be of type Integer

Update Post

URI: */ts/api/v1/post/update*

Details: Request body must contain a Post object

Updates existing Post with data in body of sent Post

Id of existing Post must be included in request body

Delete Post by ID

URI: */ts/api/v1/post/delete/{id}*

Details: Deletes an existing Post object by given Id

Path parameter must be of type Integer