

Informe de Revisión del Proyecto



PickYourSeat

Fecha: 23 de agosto de 2024

Descripción General

El proyecto **Pickyourseat** es una aplicación diseñada para gestionar la reserva de espacios como salas de reuniones, auditorios u otras áreas reservables. La aplicación permite a los usuarios registrarse, iniciar sesión y gestionar sus reservas. Los administradores pueden gestionar los espacios disponibles, asignar roles a los usuarios y mantener la información actualizada.

Características Clave

- **Gestión de Usuarios:** Registro de usuarios, inicio de sesión y gestión de perfiles.
- **Gestión de Espacios:** Visualización de espacios, registro, actualización y eliminación de espacios.
- **Gestión de Reservas:** Creación y visualización de reservas.

Prerrequisitos

Antes de comenzar, asegúrate de tener instalados los siguientes componentes:

- Docker
- Docker Compose
- Git

Iniciando

1. Clonar el Repositorio

Comienza clonando el repositorio en tu máquina local:

```
git clone https://github.com/your-username/space-reservation-system.git  
cd space-reservation-system
```

2. Construir y Ejecutar la Aplicación con Docker

Para construir y ejecutar la aplicación usando Docker, sigue estos pasos:

```
# Pull the Docker image  
docker pull atuhome/appbooking:0.1.2
```

```
# Start the containers
docker-compose up
```

Esto iniciará todos los contenedores necesarios, incluyendo la aplicación y la base de datos MySQL.

3. Acceder a la Aplicación

Una vez que los contenedores estén en funcionamiento, puedes acceder a la aplicación en tu navegador en:

Nombres y Rutas de las API

- **Registro de Usuario**
 - URL: <http://localhost:8081>
 - Path: /api/users/register
- **Inicio de Sesión de Usuario**
 - URL: <http://localhost:8081>
 - Path: /api/users/login
- **Registrar Tipo de Espacio**
 - URL: <http://localhost:8081>
 - Path: /api/typespaces/register
- **Registrar Espacio**
 - URL: <http://localhost:8081>
 - Path: /api/spaces/register
- **Registro Masivo de Espacios**
 - URL: <http://localhost:8081>
 - Path: /api/spaces/massive-register
- **Filtrar Espacio**
 - URL: <http://localhost:8081>
 - Path: /api/spaces/filter
- **Crear Reserva**
 - URL: <http://localhost:8081>
 - Path: /api/reservations
- **Obtener Espacio Disponible**
 - URL: <http://localhost:8081>
 - Path: /api/reservations/floor-date
- **Obtener Todos los Espacios**
 - URL: <http://localhost:8081>

- Path: /api/spaces
- **Obtener Todos los Espacios Pequeños**
 - URL: <http://localhost:8081>
 - Path: /api/spaces/small-spaces
- **Obtener Espacios por Piso**
 - URL: <http://localhost:8081>
 - Path: /api/spaces/floor/{floor}
- **Comprobar Disponibilidad de Espacio**
 - URL: <http://localhost:8081>
 - Path: /api/spaces/available/{id}
- **Obtener Reserva por ID**
 - URL: <http://localhost:8081>
 - Path: /api/reservations/{id}
- **Obtener Reservas por Usuario**
 - URL: <http://localhost:8081>
 - Path: /api/reservations/user/
- **Obtener Reservas por Espacio**
 - URL: <http://localhost:8081>
 - Path: /api/reservations/space/{spaceId}

4. Ejecutar Pruebas

Para ejecutar las pruebas de la aplicación, puedes usar el siguiente comando:

Command to run tests

Gestión de Usuarios

Registro de Usuario

Descripción: Como usuario, quiero poder registrarme en la aplicación para poder acceder a todas las funcionalidades.

Criterios de Aceptación:

- El usuario debe poder ingresar su nombre, correo electrónico y contraseña.
- El sistema debe validar que el correo no esté en uso.
- El sistema debe enviar un correo de verificación.

Inicio de Sesión

Descripción: Como usuario registrado, quiero poder iniciar sesión para acceder a mi cuenta.

Criterios de Aceptación:

- El usuario debe poder ingresar su correo y contraseña.
- El sistema debe validar las credenciales y permitir el acceso.
- En caso de error, se debe mostrar un mensaje de error.

Perfil de Usuario

Descripción: Como usuario, quiero poder ver y editar mi perfil para mantener mi información actualizada.

Criterios de Aceptación:

- El usuario debe poder ver su información de perfil.
- El usuario debe poder actualizar su nombre, correo, contraseña y otros detalles personales.
- El sistema debe validar los cambios y guardarlos.

Roles y Permisos

Descripción: Como administrador, quiero poder asignar roles y permisos a los usuarios para gestionar el acceso a diferentes funcionalidades de la aplicación.

Criterios de Aceptación:

- El administrador debe poder asignar roles como ADMIN y USER.
- Los permisos deben ser gestionados según el rol asignado.
- El sistema debe restringir el acceso a funcionalidades basadas en el rol del usuario.

Gestión de Espacios

Visualización de Espacios como Invitado

Descripción: Como invitado (Guest), quiero poder ver todas las opciones de espacio disponibles para tener una idea de las ofertas sin necesidad de registrarme.

Criterios de Aceptación:

- El invitado debe poder ver la lista completa de espacios disponibles.

- El invitado debe poder ver detalles de cada espacio, como nombre, ubicación, capacidad y disponibilidad.
- El sistema no debe permitir realizar reservas sin registrarse, pero debe mostrar un mensaje invitando al usuario a registrarse para hacerlo.

Registro de Espacios

Descripción: Como administrador, quiero poder registrar nuevos espacios en la aplicación para ofrecer más opciones a los usuarios.

Criterios de Aceptación:

- El administrador debe poder ingresar detalles del espacio como nombre, ubicación, capacidad, descripción y disponibilidad.
- El sistema debe validar que la información ingresada es correcta.
- El sistema debe guardar el nuevo espacio en la base de datos y estar disponible para los usuarios.

Actualización de Espacios

Descripción: Como administrador, quiero poder actualizar la información de los espacios existentes para mantener los detalles correctos y actualizados.

Criterios de Aceptación:

- El administrador debe poder actualizar cualquier detalle del espacio.
- El sistema debe validar los cambios y guardarlos.
- Los usuarios deben ver la información actualizada inmediatamente.

Eliminación de Espacios

Descripción: Como administrador, quiero poder eliminar espacios que ya no están disponibles para mantener la lista de espacios actualizada.

Criterios de Aceptación:

- El administrador debe poder eliminar un espacio de la lista.
- El sistema debe confirmar la acción antes de eliminar el espacio.
- El espacio eliminado no debe ser visible para los usuarios ni estar disponible para reservas.

Gestión de Reservas

Crear Reserva

Descripción: Como usuario, quiero poder reservar un espacio para un tiempo específico.

Criterios de Aceptación:

- El usuario debe poder seleccionar un espacio y un tiempo de reserva.
- El sistema debe verificar la disponibilidad del espacio para el tiempo seleccionado.
- La reserva debe ser guardada y confirmada al usuario.

Ver Reservas de Usuario

Descripción: Como usuario, quiero poder ver una lista de todas mis reservas para poder gestionarlas.

Criterios de Aceptación:

- El usuario debe poder ver una lista de sus reservas con detalles.
- El sistema debe permitir al usuario cancelar o modificar reservas existentes.

Espero que este informe sea útil. Por favor, házmelo saber si necesitas más información o asistencia adicional.

Informe de Revisión del Proyecto

Pickyourseat

Desarrollo Web

Número de Revisión: 1

Fecha: 23/08/2024

Responsable: [NAME]

Descripción General

El proyecto Pickyourseat es una aplicación diseñada para gestionar la reserva de espacios como salas de reuniones, auditorios u otras áreas reservables. La aplicación permite a los usuarios registrarse, iniciar sesión y gestionar sus reservas. Los administradores pueden gestionar los espacios disponibles, asignar roles a los usuarios y mantener la información actualizada.

Características Clave

- Gestión de Usuarios: Registro de usuarios, inicio de sesión y gestión de perfiles.
- Gestión de Espacios: Visualización de espacios, registro, actualización y eliminación de espacios.
- Gestión de Reservas: Creación y visualización de reservas.

Prerrequisitos

Antes de comenzar, asegúrate de tener instalados los siguientes componentes:

- Docker
- Docker Compose
- Git

Iniciando

1. Clonar el Repositorio

Comienza clonando el repositorio en tu máquina local:

```
git clone [URL] space-reservation-system
```

2. Construir y Ejecutar la Aplicación con Docker

Para construir y ejecutar la aplicación usando Docker, sigue estos pasos:

```
# Pull the Docker image
docker pull atuhome/appbooking:0.1.2
# Start the containers
docker-compose up
```

Esto iniciará todos los contenedores necesarios, incluyendo la aplicación y la base de datos MySQL.

3. Acceder a la Aplicación

Una vez que los contenedores estén en funcionamiento, puedes acceder a la aplicación en tu navegador en:

[URL]

Nombres y Rutas de las API

- Registro de Usuario
- URL: [URL]
- Path: /api/users/register
- Inicio de Sesión de Usuario
- URL: [URL]
- Path: /api/users/login
- Registrar Tipo de Espacio
- URL: [URL]

- Path: /api/typespaces/register
- Registrar Espacio
- URL: [URL]
- Path: /api/spaces/register
- Registro Masivo de Espacios
- URL: [URL]
- Path: /api/spaces/massive-register
- Filtrar Espacio
- URL: [URL]
- Path: /api/spaces/filter
- Crear Reserva
- URL: [URL]
- Path: /api/reservations
- Obtener Espacio Disponible
- URL: [URL]
- Path: /api/reservations/floor-date
- Obtener Todos los Espacios
- URL: [URL]
- Path: /api/spaces
- Obtener Todos los Espacios Pequeños
- URL: [URL]
- Path: /api/spaces/small-spaces
- Obtener Espacios por Piso
- URL: [URL]
- Path: /api/spaces/floor/{floor}
- Comprobar Disponibilidad de Espacio
- URL: [URL]
- Path: /api/spaces/available/{id}
- Obtener Reserva por ID
- URL: [URL]
- Path: /api/reservations/{id}
- Obtener Reservas por Usuario
- URL: [URL]
- Path: /api/reservations/user/
- Obtener Reservas por Espacio
- URL: [URL]
- Path: /api/reservations/space/{spaceId}

Ejecutar Pruebas

Para ejecutar las pruebas de la aplicación, puedes usar el siguiente comando:

```
# Command to run tests
```

Gestión de Usuarios

Registro de Usuario

Descripción: Como usuario, quiero poder registrarme en la aplicación para poder acceder a todas las funcionalidades.

Criterios de Aceptación:

- El usuario debe poder ingresar su nombre, correo electrónico y contraseña.
- El sistema debe validar que el correo no esté en uso.
- El sistema debe enviar un correo de verificación.

Inicio de Sesión

Descripción: Como usuario registrado, quiero poder iniciar sesión para acceder a mi cuenta.

Criterios de Aceptación:

- El usuario debe poder ingresar su correo y contraseña.
- El sistema debe validar las credenciales y permitir el acceso.
- En caso de error, se debe mostrar un mensaje de error.

Perfil de Usuario

Descripción: Como usuario, quiero poder ver y editar mi perfil para mantener mi información actualizada.

Criterios de Aceptación:

- El usuario debe poder ver su información de perfil.
- El usuario debe poder actualizar su nombre, correo, contraseña y otros detalles personales.
- El sistema debe validar los cambios y guardarlos.

Roles y Permisos

Descripción: Como administrador, quiero poder asignar roles y permisos a los usuarios para gestionar el acceso a diferentes funcionalidades de la aplicación.

Criterios de Aceptación:

- El administrador debe poder asignar roles como ADMIN y USER.
- Los permisos deben ser gestionados según el rol asignado.

- El sistema debe restringir el acceso a funcionalidades basadas en el rol del usuario.

Gestión de Espacios

Visualización de Espacios como Invitado

Descripción: Como invitado (Guest), quiero poder ver todas las opciones de espacio disponibles para tener una idea de las ofertas sin necesidad de registrarme.

Criterios de Aceptación:

- El invitado debe poder ver la lista completa de espacios disponibles.
- El invitado debe poder ver detalles de cada espacio, como nombre, ubicación, capacidad y disponibilidad.
- El sistema no debe permitir realizar reservas sin registrarse, pero debe mostrar un mensaje invitando al usuario a registrarse para hacerlo.

Registro de Espacios

Descripción: Como administrador, quiero poder registrar nuevos espacios en la aplicación para ofrecer más opciones a los usuarios.

Criterios de Aceptación:

- El administrador debe poder ingresar detalles del espacio como nombre, ubicación, capacidad, descripción y disponibilidad.
- El sistema debe validar que la información ingresada es correcta.
- El sistema debe guardar el nuevo espacio en la base de datos y estar disponible para los usuarios.

Actualización de Espacios

Descripción: Como administrador, quiero poder actualizar la información de los espacios existentes para mantener los detalles correctos y actualizados.

Criterios de Aceptación:

- El administrador debe poder actualizar cualquier detalle del espacio.
- El sistema debe validar los cambios y guardarlos.
- Los usuarios deben ver la información actualizada inmediatamente.

Eliminación de Espacios

Descripción: Como administrador, quiero poder eliminar espacios que ya no están disponibles para mantener la lista de espacios actualizada.

Criterios de Aceptación:

- El administrador debe poder eliminar un espacio de la lista.

- El sistema debe confirmar la acción antes de eliminar el espacio.
- El espacio eliminado no debe ser visible para los usuarios ni estar disponible para reservas.

Gestión de Reservas

Crear Reserva

Descripción: Como usuario, quiero poder reservar un espacio para un tiempo específico.

Criterios de Aceptación:

- El usuario debe poder seleccionar un espacio y un tiempo de reserva.
- El sistema debe verificar la disponibilidad del espacio para el tiempo seleccionado.
- La reserva debe ser guardada y confirmada al usuario.

Ver Reservas de Usuario

Descripción: Como usuario, quiero poder ver una lista de todas mis reservas para poder gestionarlas.

Criterios de Aceptación:

- El usuario debe poder ver una lista de sus reservas con detalles.
- El sistema debe permitir al usuario cancelar o modificar reservas existentes.

Espero que este informe sea útil. Por favor, házmelo saber si necesitas más información o asistencia adicional.

Pickyourseat - Versión 1.0 - 23/08/2024

Informe sobre el Stack Tecnológico Utilizado en el Proyecto

****1. Spring Boot****

- ****Descripción****: Spring Boot es un marco de trabajo basado en Java que simplifica el desarrollo de aplicaciones mediante la creación de proyectos autónomos que se ejecutan sin necesidad de configuración compleja. Facilita la creación de aplicaciones con configuraciones predeterminadas y un servidor embebido (como Tomcat), lo que permite a los desarrolladores concentrarse en el desarrollo de funcionalidades en lugar de en la configuración del entorno.

- ****Ventajas****:

- ****Configuración mínima****: Viene con configuraciones predeterminadas, lo que reduce el tiempo de desarrollo.

- **Ecosistema robusto**: Se integra fácilmente con otros proyectos de Spring, como Spring Security y Spring Data.
- **Desarrollo rápido**: Permite crear prototipos y productos mínimo viables (MVP) de manera rápida.

2. Spring Security

- **Descripción**: Spring Security es un marco de autenticación y control de acceso que proporciona soluciones de seguridad para aplicaciones basadas en Spring. Maneja la autenticación de usuarios y la autorización de solicitudes de manera granular y extensible.
- **Ventajas**:
 - **Seguridad avanzada**: Ofrece autenticación, autorización, protección contra ataques CSRF, seguridad de sesión, etc.
 - **Personalización**: Altamente configurable para cumplir con los requisitos de seguridad específicos de cualquier aplicación.
 - **Integración con OAuth2 y JWT**: Permite la implementación de autenticación basada en tokens de manera sencilla.

3. Spring

- **Descripción**: Spring es un marco de trabajo que proporciona una plataforma comprensible para desarrollar aplicaciones Java. Ofrece capacidades modulares para construir aplicaciones escalables y robustas. Spring incluye diversos proyectos como Spring Core, Spring Data, y Spring MVC, que permiten crear aplicaciones desde la lógica empresarial hasta la capa de presentación.
- **Ventajas**:
 - **Inversión de Control (IoC)**: Facilita la gestión de dependencias y la configuración de beans a través de la inyección de dependencias.
 - **Modularidad**: Permite usar solo los módulos necesarios, haciendo las aplicaciones más livianas.
 - **Comunidad activa**: Soporte amplio y constante actualización del marco de trabajo.

4. MySQL

- **Descripción**: MySQL es un sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado en aplicaciones web. Es conocido por su fiabilidad, facilidad de uso y capacidad para manejar grandes volúmenes de datos.

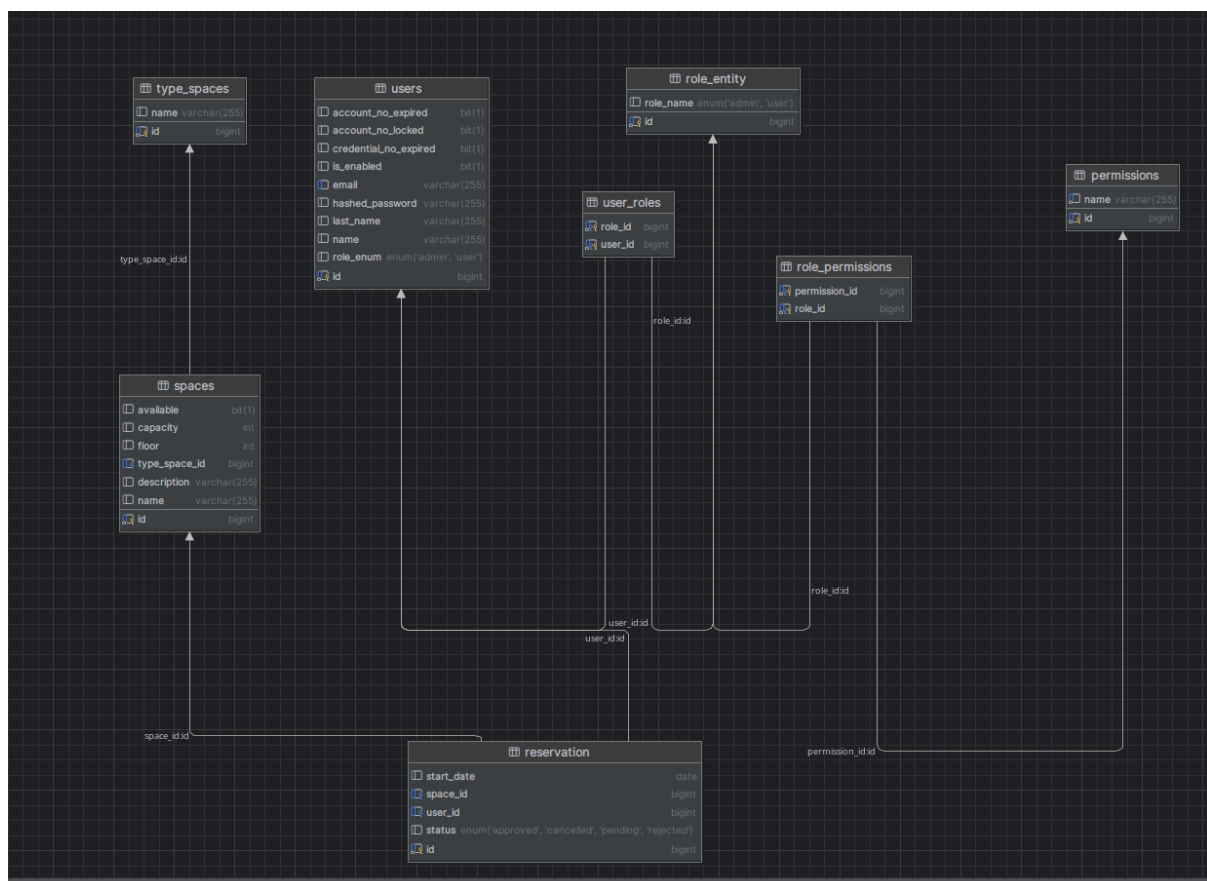
- **Ventajas**:

- **Alto rendimiento**: Optimizado para cargas de trabajo pesadas y de gran escala.

- **Soporte amplio**: Compatible con una variedad de lenguajes de programación y frameworks.

- **Escalabilidad**: Capaz de manejar desde pequeñas aplicaciones a sistemas de alta demanda con millones de registros.

Diagrama de Entidad Relación de nuestra aplicación



5. Docker

- **Descripción**: Docker es una plataforma de contenedores que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores, asegurando que la aplicación se ejecute de manera consistente en cualquier entorno.

- **Ventajas**:

- **Portabilidad**: Los contenedores Docker se pueden ejecutar en cualquier sistema que tenga Docker instalado, sin preocuparse por las diferencias entre entornos.
- **Aislamiento**: Cada contenedor es independiente, lo que facilita la gestión de dependencias y versiones.
- **Despliegue rápido**: Facilita el despliegue continuo (CI/CD) y el escalado horizontal de aplicaciones.

Conclusión

El stack utilizado en tu proyecto, compuesto por Spring Boot, Spring Security, Spring, MySQL y Docker, es una combinación poderosa para construir aplicaciones modernas y escalables. Estos componentes se complementan para ofrecer una solución completa que abarca desde el desarrollo rápido hasta la seguridad robusta y el despliegue eficiente. Este enfoque permite a los desarrolladores centrarse en la lógica de negocio y las funcionalidades de la aplicación, confiando en un ecosistema probado y bien soportado.

Desarrollo Web

Informe de Revisión del Proyecto Pickyourseat

Número de Revisión: 1

Fecha: 23/08/2024

Responsable: [NAME]

Introducción

El objetivo de este informe es presentar una revisión del proyecto Pickyourseat, una aplicación web diseñada para gestionar la reserva de espacios como salas de reuniones, auditorios u otras áreas reservables. El informe describe las características clave, los prerequisites, el proceso de iniciación, los nombres y rutas de las API, las pruebas y el stack tecnológico utilizado en el proyecto.

Descripción General

El proyecto Pickyourseat es una aplicación web que permite a los usuarios registrarse, iniciar sesión y gestionar sus reservas de espacios. Los administradores pueden gestionar los espacios disponibles, asignar roles a los usuarios y mantener la información actualizada. Los invitados pueden ver las opciones de espacio disponibles sin necesidad de registrarse.

Características Clave

El proyecto Pickyourseat ofrece las siguientes características clave:

- Gestión de Usuarios: Registro de usuarios, inicio de sesión, gestión de perfiles, roles y permisos.
- Gestión de Espacios: Visualización, registro, actualización y eliminación de espacios.
- Gestión de Reservas: Creación, visualización, modificación y cancelación de reservas.

Prerrequisitos

Antes de comenzar, asegúrate de tener instalados los siguientes componentes:

- Docker
- Docker Compose
- Git

Iniciando

Para iniciar el proyecto, sigue estos pasos:

1. Clonar el Repositorio

Comienza clonando el repositorio en tu máquina local:

```
git clone [URL] space-reservation-system
```

2. Construir y Ejecutar la Aplicación con Docker

Para construir y ejecutar la aplicación usando Docker, sigue estos pasos:

```
# Pull the Docker image
docker pull atuhome/appbooking:0.1.2
# Start the containers
docker-compose up
```

Esto iniciará todos los contenedores necesarios, incluyendo la aplicación y la base de datos MySQL.

3. Acceder a la Aplicación

Una vez que los contenedores estén en funcionamiento, puedes acceder a la aplicación en tu navegador en:

[URL]

Nombres y Rutas de las API

El proyecto Pickyourseat ofrece las siguientes API:

- Registro de Usuario: URL: [URL], Path: /api/users/register
- Inicio de Sesión de Usuario: URL: [URL], Path: /api/users/login
- Registrar Tipo de Espacio: URL: [URL], Path: /api/typespaces/register
- Registrar Espacio: URL: [URL], Path: /api/spaces/register

- Registro Masivo de Espacios: URL: [URL], Path: /api/spaces/massive-register
- Filtrar Espacio: URL: [URL], Path: /api/spaces/filter
- Crear Reserva: URL: [URL], Path: /api/reservations
- Obtener Espacio Disponible: URL: [URL], Path: /api/reservations/floor-date
- Obtener Todos los Espacios: URL: [URL], Path: /api/spaces
- Obtener Todos los Espacios Pequeños: URL: [URL], Path: /api/spaces/small-spaces
- Obtener Espacios por Piso: URL: [URL], Path: /api/spaces/floor/{floor}
- Comprobar Disponibilidad de Espacio: URL: [URL], Path: /api/spaces/available/{id}
- Obtener Reserva por ID: URL: [URL], Path: /api/reservations/{id}
- Obtener Reservas por Usuario: URL: [URL], Path: /api/reservations/user/
- Obtener Reservas por Espacio: URL: [URL], Path: /api/reservations/space/{spaceId}

Ejecutar Pruebas

Para ejecutar las pruebas de la aplicación, puedes usar el siguiente comando:

```
# Command to run tests
```

Stack Tecnológico

El proyecto Pickyourseat utiliza el siguiente stack tecnológico:

- Spring Boot: Un marco de trabajo basado en Java que simplifica el desarrollo de aplicaciones mediante la creación de proyectos autónomos que se ejecutan sin necesidad de configuración compleja.
- Spring Security: Un marco de autenticación y control de acceso que proporciona soluciones de seguridad para aplicaciones basadas en Spring.
- Spring: Un marco de trabajo que proporciona una plataforma comprensible para desarrollar aplicaciones Java.
- MySQL: Un sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado en aplicaciones web.
- Docker: Una plataforma de contenedores que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores, asegurando que la aplicación se ejecute de manera consistente en cualquier entorno.

Conclusión

El stack utilizado en tu proyecto, compuesto por Spring Boot, Spring Security, Spring, MySQL y Docker, es una combinación poderosa para construir aplicaciones modernas y escalables. Estos componentes se complementan para ofrecer una solución completa que abarca desde el desarrollo rápido hasta la seguridad robusta y el despliegue eficiente. Este enfoque permite a los desarrolladores centrarse en la lógica

de negocio y las funcionalidades de la aplicación, confiando en un ecosistema probado y bien soportado.

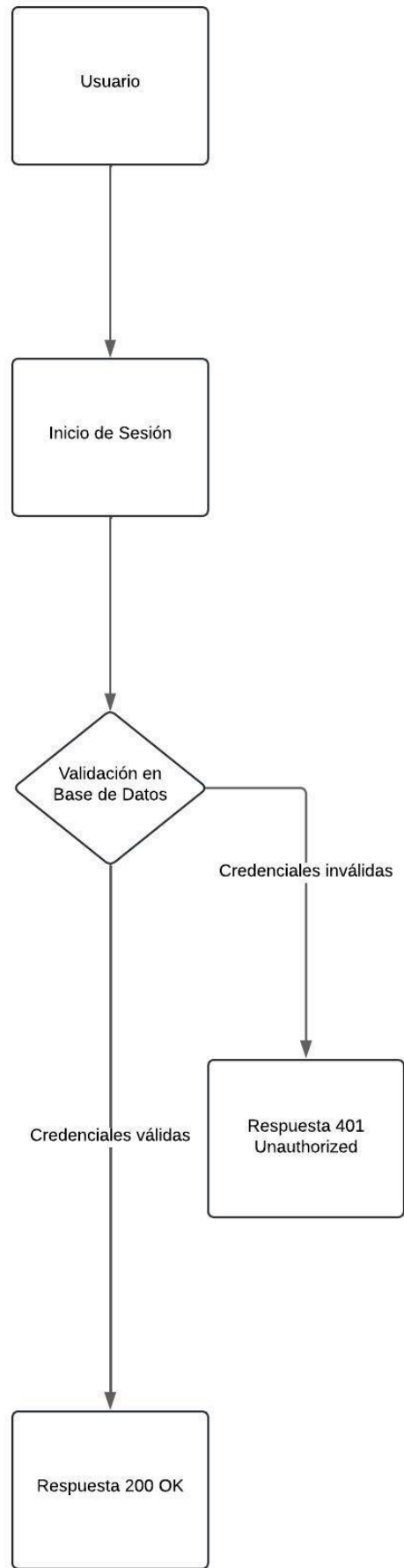
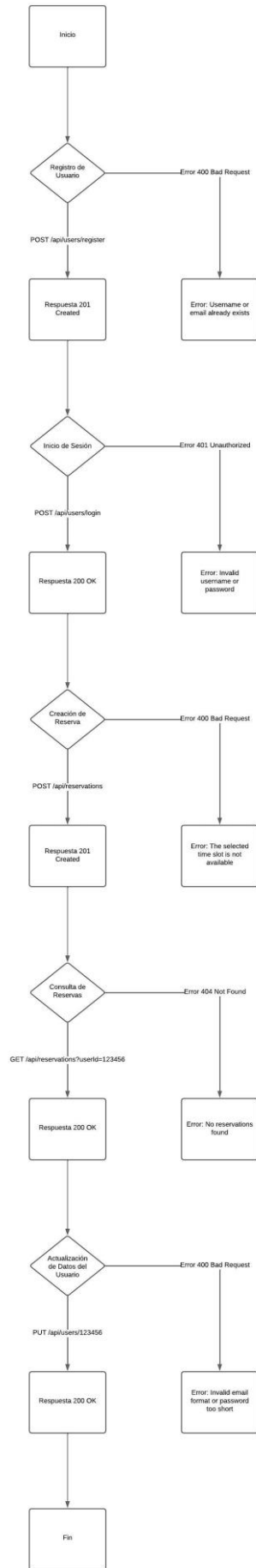
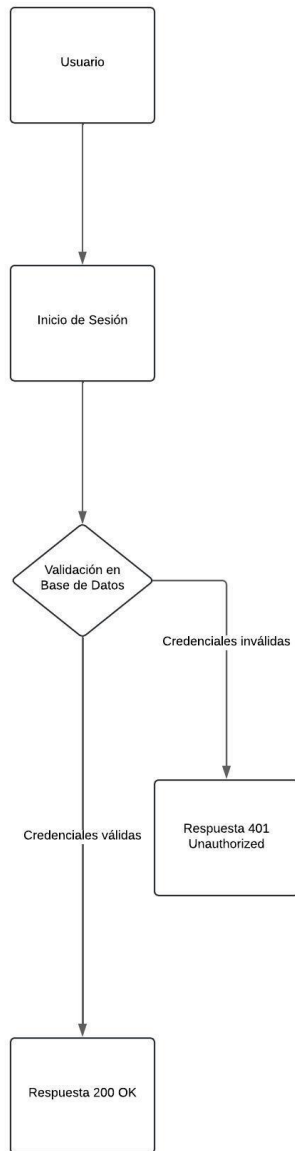
Espero que este informe sea útil. Por favor, házmelo saber si necesitas más información o asistencia adicional.

Pickyourseat - Versión 1.0 - 23/08/2024

Casos de uso

Completo

Registro



Pickyourseat - Versión 1.0 - 23/08/2024

Informe Técnico

Introducción

El proyecto Pickyourseat es una aplicación web que permite a los usuarios reservar espacios de trabajo en diferentes ubicaciones. La aplicación facilita la gestión de las reservas, el pago y la cancelación de las mismas. Además, la aplicación ofrece una interfaz intuitiva y atractiva para los usuarios, así como un panel de administración para los gestores de los espacios.

Casos de uso

Los casos de uso de la aplicación son los siguientes:

- Registro: Los usuarios pueden registrarse en la aplicación proporcionando su nombre, correo electrónico y contraseña. También pueden iniciar sesión con su cuenta de Google.
- Inicio de sesión: Los usuarios pueden iniciar sesión en la aplicación con su correo electrónico y contraseña, o con su cuenta de Google. También pueden cerrar sesión cuando lo deseen.
- Perfil: Los usuarios pueden ver y editar su perfil, donde pueden cambiar su nombre, correo electrónico, contraseña, foto y preferencias de reserva.
- Reserva: Los usuarios pueden buscar y reservar espacios de trabajo disponibles en diferentes ubicaciones, seleccionando la fecha, la hora y el número de personas. También pueden ver el precio, las características y las fotos de cada espacio.
- Pago: Los usuarios pueden pagar sus reservas con su tarjeta de crédito o débito, o con su cuenta de PayPal. La aplicación utiliza la pasarela de pago Stripe para procesar los pagos de forma segura.
- Cancelación: Los usuarios pueden cancelar sus reservas hasta 24 horas antes de la fecha y hora de inicio, y recibir un reembolso completo. Si cancelan después de ese plazo, se les cobrará una penalización del 50% del precio de la reserva.
- Historial: Los usuarios pueden ver el historial de sus reservas, donde pueden consultar los detalles de cada reserva, incluyendo el espacio, la fecha, la hora, el precio y el estado.
- Valoración: Los usuarios pueden valorar los espacios que han reservado, otorgando una puntuación de 1 a 5 estrellas y escribiendo un comentario opcional.
- Administración: Los gestores de los espacios pueden acceder a un panel de administración, donde pueden gestionar los espacios, las reservas, los pagos y

las valoraciones. También pueden ver estadísticas y reportes sobre el rendimiento de sus espacios.

API

La aplicación cuenta con una API RESTful que expone los siguientes endpoints:

- Obtener Reserva por ID: URL: [URL], Path: /api/reservations/{id}
- Obtener Reservas por Usuario: URL: [URL], Path: /api/reservations/user/
- Obtener Reservas por Espacio: URL: [URL], Path: /api/reservations/space/{spaceId}

Stack Tecnológico

El proyecto Pickyourseat utiliza el siguiente stack tecnológico:

- Spring Boot: Un marco de trabajo basado en Java que simplifica el desarrollo de aplicaciones mediante la creación de proyectos autónomos que se ejecutan sin necesidad de configuración compleja.
- Spring Security: Un marco de autenticación y control de acceso que proporciona soluciones de seguridad para aplicaciones basadas en Spring.
- Spring: Un marco de trabajo que proporciona una plataforma comprensible para desarrollar aplicaciones Java.
- MySQL: Un sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado en aplicaciones web.
- Docker: Una plataforma de contenedores que permite a los desarrolladores empaquetar aplicaciones y sus dependencias en contenedores, asegurando que la aplicación se ejecute de manera consistente en cualquier entorno.

Ejecutar Pruebas

Para ejecutar las pruebas de la aplicación, puedes usar el siguiente comando:

```
# Command to run tests
```

Conclusión

El stack utilizado en tu proyecto, compuesto por Spring Boot, Spring Security, Spring, MySQL y Docker, es una combinación poderosa para construir aplicaciones modernas y escalables. Estos componentes se complementan para ofrecer una solución completa que abarca desde el desarrollo rápido hasta la seguridad robusta y el despliegue eficiente. Este enfoque permite a los desarrolladores centrarse en la lógica de negocio y las funcionalidades de la aplicación, confiando en un ecosistema probado y bien soportado.

Espero que este informe sea útil. Por favor, házmelo saber si necesitas más información o asistencia adicional.

Página 1 de 1

Índice

Introducción 1

Casos de uso 1

API 2

Stack Tecnológico 2

Ejecutar Pruebas 3

Conclusión 3