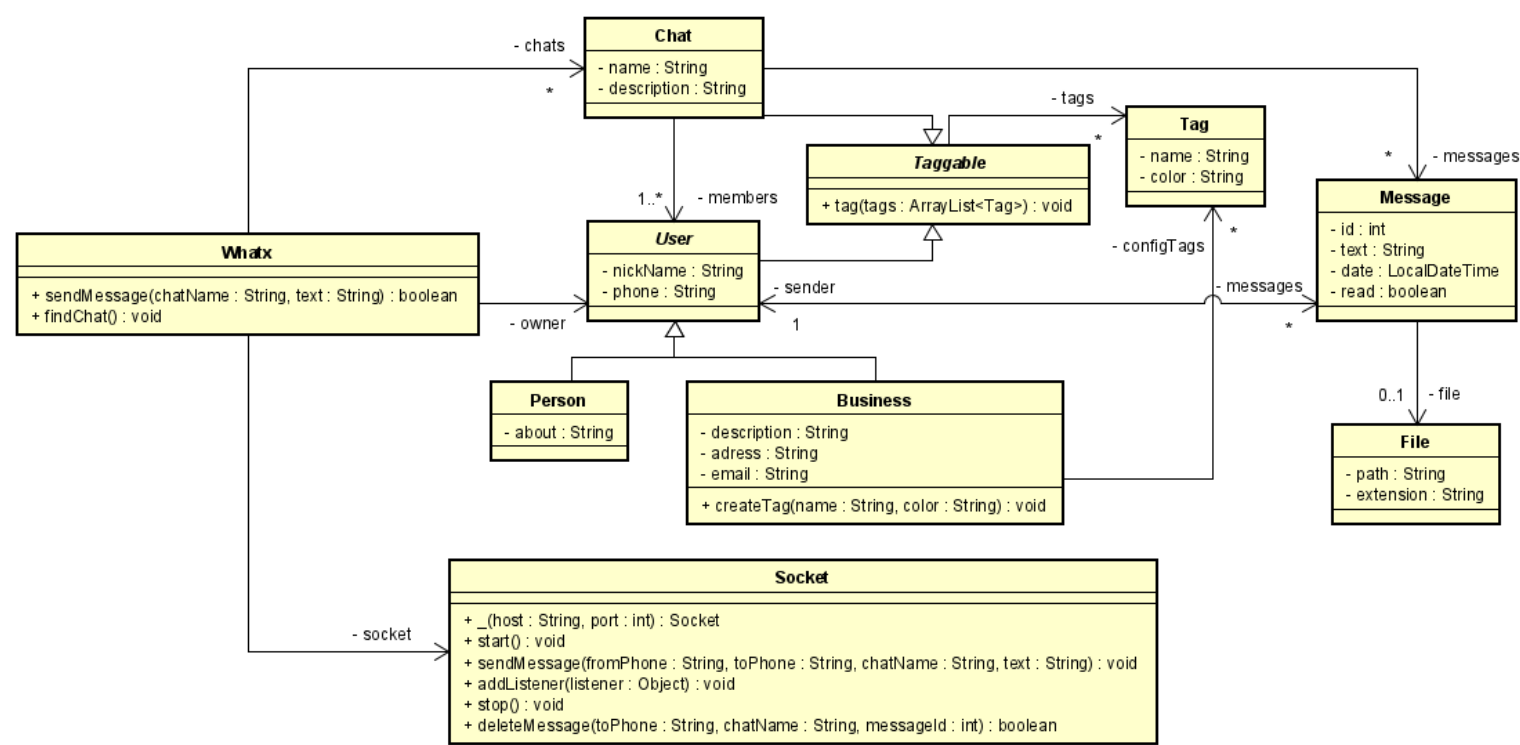


Whatx

Whatx ha decidido extender sus tipos de usuarios a: personas y empresas. Para las personas, Whatx permite enviar y recibir mensajes. A nivel empresarial, Whatx provee funcionalidades adicionales:

- Etiquetar: Le permite configurar diferentes etiquetas para agrupar elementos relacionados con temas en común (Ej: proveedores, clientes premium, entre otras). Se podrán relacionar chats y otros usuarios a las etiquetas definidas, teniendo en cuenta:
 - De cada etiqueta se define: nombre y color.
 - No pueden existir etiquetas con el mismo nombre (criterio de unicidad).
 - Cada empresa configura sus propias etiquetas.
- Mensajes automáticos: Configura mensajes por defecto.



(Todos los contenedores son ArrayList)

Class Taggable	
void	tag (ArrayList<Tag> tags) Associate elements to business tags NO_TAGS tags list is empty. TAGGED if the element is already associated with some tag. (Tag until exception is found) NO_MEMBERS if the element is chat, and it doesn't have any members. INCOMPLETE_INFORMATION if any tags don't have complete information (name or color empty).

Class LocalDate - Method Summary	
LocalDate is an immutable date-time object that represents a date, often viewed as year-month-day.	
boolean	isEqual s(LocalDate other) Checks if this date is equal to the specified date.
boolean	isAfter (LocalDate other) Checks if this date is after the specified date.
boolean	isBefore (LocalDate other) Checks if this date is equal to the specified date.
Static LocalDate	now () Obtains the current date from the system clock.

I. (35%) PREPARANDO ETIQUETAS

Implemente los métodos necesarios para cumplir con los requisitos asociados a la clase *Taggable*. Esta clase permite asociar elementos (chat y usuarios) a etiquetas definidas por usuarios empresariales. **Nota:** Tener presente que al etiquetar un chat se deben etiquetar todos sus miembros. No olvide MDD

MDD

1. Escriba la especificación (documentación + encabezado) del método
2. Construya el(los) diagrama(s) de secuencia (adicione el manejo de excepciones con otro color)
3. Actualice el diagrama de clases con los nuevos elementos
4. Escriba el código de las clases involucradas en el método (encabezado y atributos) y escriba el código necesario para implementar el método especificado.

II. (20%) DISEÑANDO

Diseñe el siguiente método

MDD

1. Estudie la especificación (documentación + encabezado) del método y las características de *Whatx*
2. Realice el diagrama de secuencia (adicione el manejo de excepciones con otro color)
3. Actualice el diagrama de clases con los nuevos elementos.

En *Whatx*

```
public ArrayList<User> findUsersByTagName(String tagName, LocalDate startDate, LocalDate endDate)
```

Gets users associated with a specific tag. Users have sent messages in a date range.

Parameters:

tagName - Tag's name
startDate - Range start date
endDate - Range end date

Returns:

Users list associated with the tag.

Throws:

- WhatxException - NO_TAG** There is no tag with that name
- **FUTURE_DATES** StartDate and endDate are future dates. (After today)
- **START_BEFORE_END** StartDate is before endDate

III. (25%) EXTENDIENDO

Whatx desea adicionar para los usuarios empresariales la funcionalidad de definir mensajes automáticos con diferentes tipos de archivos: documento, contactos y videos. Considere las siguientes reglas de negocio:

- Cada mensaje automático se puede etiquetar.
- De los tipos de archivo: documento y video, se conoce su formato y peso.
- Los videos pueden tener una duración de hasta 3 minutos.
- De los contactos se tiene la siguiente información: nombre y teléfono.

MDD

1. Realice los cambios necesarios en el diagrama de clases.
2. Implemente dichos cambios, solo estructurales definidos en el diagrama de clases.
3. Analice los diseños anteriores y explique los cambios adicionales a realizar.
4. Considerando el segundo principio SOLID ¿Qué es lo positivo y/o negativo del diseño?
5. Modifique o realice los nuevos diseños

IV. (20%) Conceptos

1. ¿Qué es una clase abstracta? ¿Cuál podría ser una diferencia con una interfaz?
2. ¿Qué es un método *default* en una interfaz? Indique una ventaja de su definición al extender un modelo.