@startuml

skinparam classAttributeIconSize 0

package "com.company.chat.application.service" {

class ChatService {

- chatMediator : ChatMediator

+ registerMessage(messageRequest : MessageRequest) : void

+ addUser(user : ChatUser) : void

}

}

package "com.company.chat.domain.mediator" {

interface ChatMediator {

+ addUser(user : ChatUser) : void

+ removeUser(user : ChatUser) : void

+ sendMessage(message : Message, sender : ChatUser) : void

}

class ChatRoom {

- users : List<ChatUser>

- messageHistory : List<Message>

- jpaMessageAdapter : JpaMessageAdapter

+ addUser(user : ChatUser) : void

+ removeUser(user : ChatUser) : void

+ sendMessage(message : Message, sender : ChatUser) : void

- notifyUsers(sender : ChatUser, message : Message) : void

+ getMessageHistory() : List<Message>

}

}

package "com.company.chat.domain.model" {

class ChatUser {

- id : String

- username : String

- email : String

- chatMediator : ChatMediator

+ setChatMediator(chatMediator : ChatMediator) : void

+ receiveMessage(message : Message) : void

}

class Message {

- sender : String

- receiver : String

- content : String

- timestamp : LocalDateTime

}

}

package "com.company.chat.dto" {

class MessageRequest {

- sender : String

- receiver : String

- content : String

+ getSender() : String

+ getReceiver() : String

+ getContent() : String

}

}

package "com.company.chat.infrastructure.persistence" {

class JpaMessageAdapter {

- messageRepository : MessageRepository

+ saveMessage(message : Message) : void

+ getAllMessages() : List<Message>

}

class MessageRepository {

+ save(entity : MessageEntity) : void

+ findAll() : List<MessageEntity>

}

class MessageEntity {

- senderId : Long

- receiverId : Long

- content : String

- timestamp : LocalDateTime

}

}

ChatService --> ChatMediator : uses

ChatRoom ..|> ChatMediator

ChatRoom --> JpaMessageAdapter : uses

ChatRoom --> ChatUser : interacts

ChatRoom --> Message : interacts

ChatUser --> ChatMediator : interacts

ChatUser --> Message : receives

JpaMessageAdapter --> MessageRepository : uses

JpaMessageAdapter --> Message : converts

JpaMessageAdapter --> MessageEntity : converts

MessageRepository --> MessageEntity : manages

@enduml