Sensor Monitoring System and  
Real-Time Notification

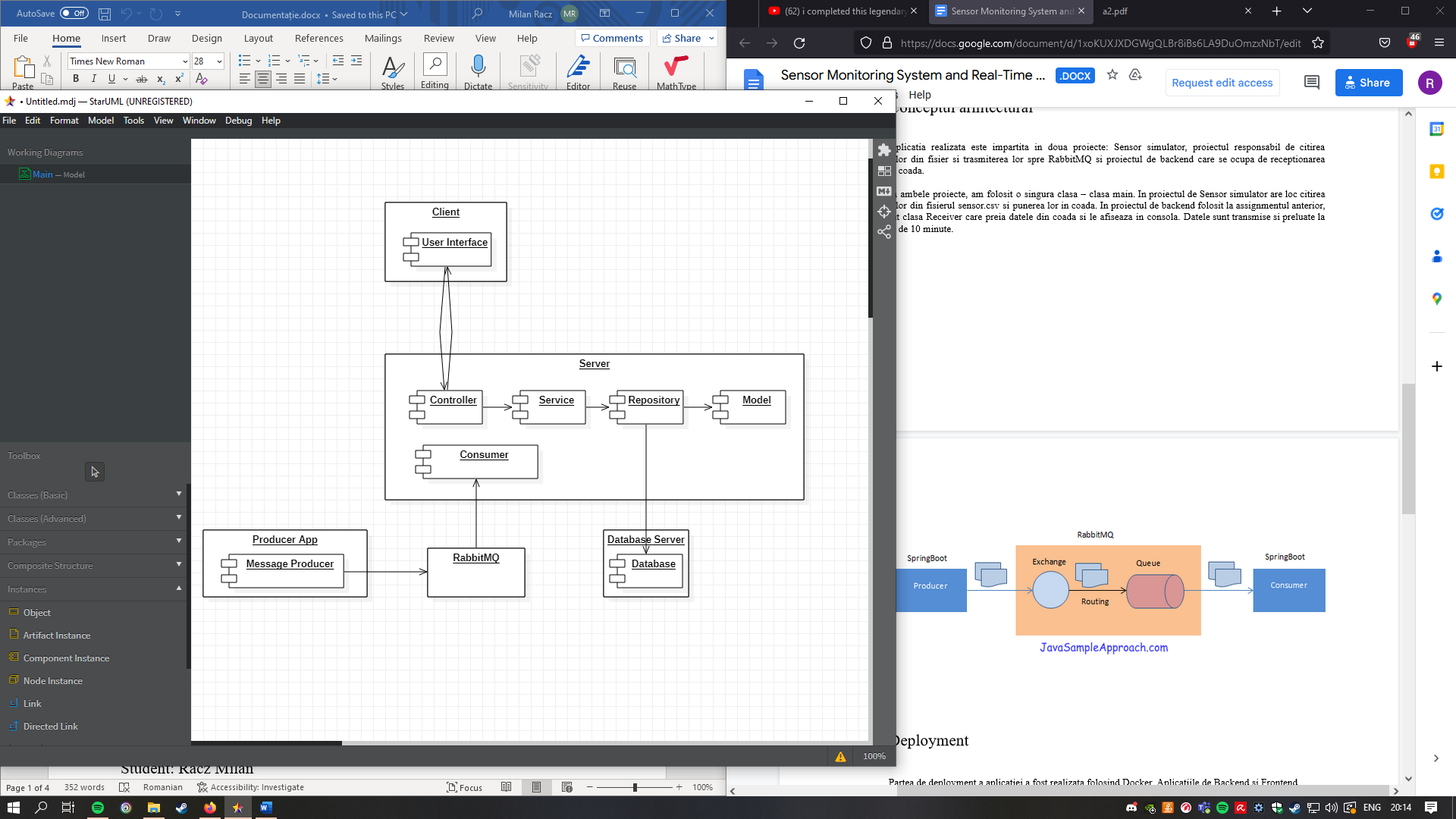
Student: Racz Milan

Grupa: 30644

# Cerințele aplicației

Trebuia implementată o aplicație care adună date de la dispozitivele inteligente de contorizare, preprocesează datele pentru a calcula energia consumată în fiecare oră și stochează datele într-o bază de date. Un modul Smart Metering Simulator va fi producătorul de mesaje. Va simula un senzor prin citirea măsurătorilor dintr-un fișier (sensor.csv - o valoare la fiecare 10 minute) și trimite date sub forma <timestamp, device\_id, measurement\_value> către Message Broker. Simulatorul de senzori ar trebui dezvoltat ca o aplicație independentă (adică, aplicație desktop). O aplicație Message Consumer va preprocesa datele pentru a calcula energia totală consumată. Dacă consumul total de energie orar calculat depășește valoarea maximă a dispozitivului inteligent, acesta notifică în mod asincron utilizatorul pe interfața sa web utilizând un Websocket.

# 2.Arhitectura conceptuală a sistemului



Aplicația este împărțită în două componente: simulatorul de senzor (care citește și trimite măsurătorile prin RabbitMQ) și aplicația realizată pentru assignment-ul anterior (care se ocupă de prelucrarea datelor primite).

# 3. Diagrama UML de Deployment

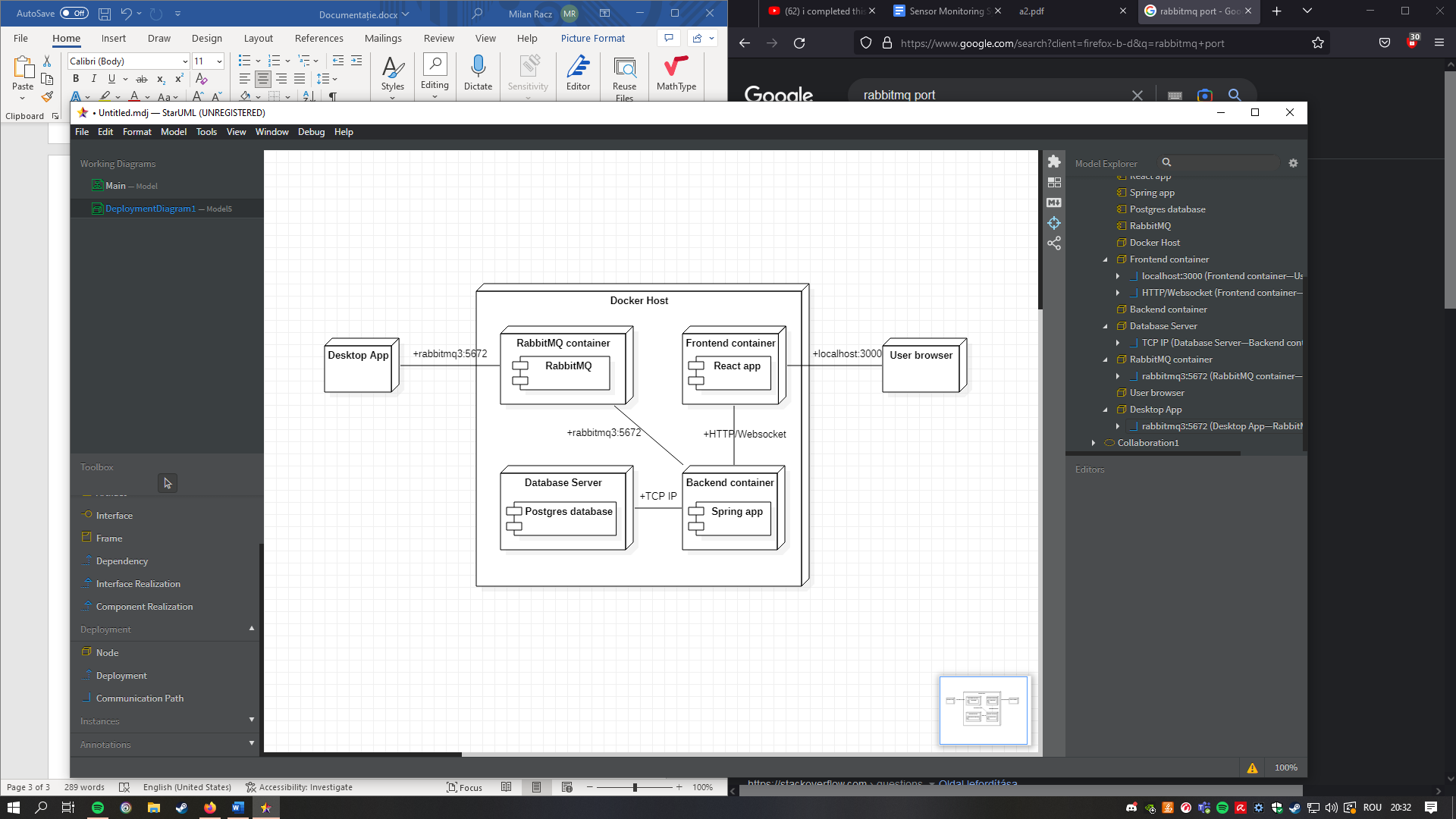


Diagrama arată comunicarea dintre componentele aplicației: partea de server, client baza de date, RabbitMQ și aplicația desktop. Backend-ul și frontend-ul comunică cu servicii REST, prin HTTP request-uri, iar datele se trimit în fișiere cu formatul .json. Backend-ul comunică cu baza de date utilizând protocolul TCP/IP. Aplicația desktop și backend-ul comunică cu RabbitMQ utilizând portul 5672. Deploymentul a fost realizat local, folosind Docker.

# 4. Readme

Pentru că deploymentul a fost făcut doar local în docker, prima dată trebuie create trei directoare pentru aplicația desktop, backend și frontend, și aceste părți trebuie clonate din repository-urile lor. După această parte imaginile celor patru părți (backend, frontend, RabbitMQ și db) trebuie create și fișierul docker-compose.yml din backend trebuie executat.