**Outsourcing**

**Konklussion**

Vi har lært eller fået bekræftet følgende:

Kommunikation er svært - især på skrift.

En outsourcet opgave skal følges stramt - 12 timer uden vi svarer en freelancer kan gøre, at han tager en anden opgave. Det er vigtigt at have udpeget en ressource der styrer forløbet frem for at vente på at en større gruppe af mennesker kan mødes og blive enige om vejen frem. Vi erkender at det kunne vi have gjort endnu tidligere i forløbet.

Der er kulturelle forskelle i hvordan vi håndterer, at der er noget vi ikke kan finde ud af - nogen går i skjul andre går til bekendelse - nogle dog først når de skulle have leveret.

Vi endte op uden produkt - havde forventet at modtage noget - måske bare i en anden kvalitet end forventet.

Hændelsesforløb:

Vi er tre grupper der sammen over to dage har stillet krav til outsourcing af webscraping af flightdata.. Svær proces at skulle beskrive kravene selvom vi egentlig godt ved hvad vi vil have retur. Især at få beskrevet så det ikke kan misforstås/mistolkes er svært.

Vi gik efter timebetaling frem for fast pris - vi baserede denne beslutning på, at vi mente at når vi var tre grupper sammen, kunne vi tåle at der kom nogle ekstra timer på til fejlhåndtering og stadigvæk være under budgettet.

Vi fik hurtigt de første svar på Freelancer - og fik via feedback lejlighed til at erkende at den opgave vi havde lagt op skulle skæres lidt ned hvis budgettet skulle holdes. Vi vedlægger opgavebeskrivelsen som bilag 2 under denne gennemgang af processen.

Vi modtog igen hurtig respons på den nye opgave og generelt var opgaven misforstået - vi bad om webscraping - de fleste tilbød både frontend og backeend.

Ud af de første mange svar vi fik udvalgte vi 7 til en kandidatliste og gik i dialog med dem med spørgsmål til opgaveforståelsen - spørgsmål vedlagt under denne gennemgang.

Vores oplevelse var herefter at det skulle vi aldrig have gjort. De få der gad besvare spørgsmålene svarede igen helt i hegnet - og en havde fået anden beskæftigelse i mellemtiden

Vi valgte derfor at lægge opgaven op på ny - denne gang på Upwork - opgave vedlagt som bilag 1.

I vores outsourcinggruppe valgte vi at have en anden tilgang til opgaven - denne gang ville vi ikke tage en kæmpe dialog. Der var en der så ud til at have forstået opgaven og ville 150 $ for at løse opgaven. Vi skrev til ham at vi gerne ville antage ham og hvilke milepæle vi ønskede, men han forsvandt og vi antog en anden fra indien denne gang via Freelancer.

Han igangsatte projektet efter Skype møde hvor han gav udtryk for at opgaven var tydeligt beskrevet og han kunne løse den. Vi var i dialog undervejs - han lovede at levere fredag den 11.12.15 - men intet kom - til gengæld skrev han søndag at han ikke kunne levere da han ikke kunne finde ud af det. Efter dialog med ham opsagde vi samarbejdet og han gav udtryk for han ingen betaling ønskede

Vi lagde opgaven op endnu engang, denne gang som ren tasteopgave hvor vi bad om flyafgange i json format - fandt hurtigt en fyr fra Bangladesh, men efter vi hyrede ham, viste det sig dog at han alligevel ikke kunne nå at løse opgaven og vi stoppede samarbejdet.

Bilag 1

Endelig opgave på Upwork:

Overall description of our project.

We are doing a project regarding fetching flight data from various airline companies and presenting them in our own airline.

We are using Java 7, AngularJS, Javascript, MySQL, Tomcat 7 and JSON. We deploy on Openshift. The twist in this project is that we are supposed to outsource some of the work.

What we want you to do - Screen scraping:

Create Java utility classes to fetch actual data from momondo.com and return it as a list of JSON objects.

We want you to scrape all direct flights from today and every day the next 3 months from:

Copenhagen, Beijing, New York (JFK), Bangkok, Dubai, Delhi, Paris, Berlin.

If there are more than 5.000 results you must return a mix that ensures that we have all cities represented in the list of JSON objects and only return maximum 5.000 JSON objects.

These classes will be integrated in our project and it is our job to manipulate and persist the JSON that your classes return in MySQL.

The application must be multithreaded and use the Callable interface.

You are supposed to ensure that the program run automatically every morning at 05.00 UTC+1. Be aware that our application will be deployed on Openshift and is supposed to run without any manual interaction.

Can you make an estimate on how many hours and which date you can start the assignment and which date you are going to finish it?

The json object are required to use this jsonSchema:

{

"title": "Airline Schema",

"type": "object",

"properties": {

"airline": {

"description": "name of the airline i.e. SAS, KLM or Singapore ",

"type": "string"

},

"date": {

"description": "The date has to be defined in ISO-8601 format and it is the departure date we want " (yyyy-MM-ddThh:mm:ss.sssZ) (the date the plane leaves the departure airport),

"type": "string"

},

"numberOfSeats": {

"description" : "Number of all the available seats on the flight",

"type": "integer",

"minimum": 0

},

"totalPrice": {

"description": " We want the price of one ticket issued in Euro ",

"type": "number",

"minimum": 0

},

"flightID": {

"description": " Unique identifier for a specific flight i.e. SK567",

"type": "string"

},

"traveltime": {

"description": " Must be issued in minutes ",

"type": "integer",

"minimum": 0

},

"destination": {

"description": "Where the flight fly to. This has to be defined in IATA-Code i.e. CPH for Copenhagen.",

"type": "string"

},

"origin": {

"description": " Where the flights fly from. This has to be defined in IATA- Code ",

"type": "string"

},

},

"required": ["airline ", " date ", "numberOfSeats", "totalPrice", "flightID", "traveltime", "destination", "origin"]

}

Example:

{

"airline": "SAS",

"flights" :[

{

"date": "2016-02-25T11:30:00.000Z",

"numberOfSeats": 212,

"totalPrice": 560,

"flightID": "SK645",

"traveltime": 480,

"destination":"CPH" ,

"origin": "JFK"

},

{

"date": "2016-08-09T22:30:00.000Z",

"numberOfSeats": 105,

"totalPrice": 680,

"flightID": "SK873",

"traveltime": 720,

"destination": "SIN",

"origin": " CPH "

}

]

}

bilag 2

Oprindelig opgave (de to sider omkring json schema og eksempel var også med i denne opgave)

Overall description of our project.

We are doing a school project regarding fetching flight data from various airline companies and presenting them in our own airline.

We are using Java 7, AngularJS, Javascript, MySQL, Tomcat 7 and JSON. We deploy on Openshift. The twist in this project is that we are supposed to outsource some of the work.

What we want you to do - Screen scraping:

Create Java utility classes to fetch actual data from 3 different airline companies and return it as a list of JSON objects.

The three companies are:

KLM = https://www.klm.com/home/gb/en

SAS = http://www.flysas.com/en/uk/

Singapore Airlines = http://www.singaporeair.com/en\_UK/dk/home

We want you to scrape all **direct** flights from today and every day the next 3 month from:

Copenhagen, Beijing, New York (JFK), Bangkok, Dubai, Delhi, Paris, Berlin.

If there are more than 5.000 results you must return a mix that ensures that we have all cities and airline companies represented in the list of JSON objects and only return maximum 5.000 JSON objects.

The application must be multithreaded and use the Callable interface.

These classes will be integrated in our project and it is our job to manipulate and persist the JSON that your classes return in MySQL.

You are supposed to ensure that the program run automatically every morning at 05.00 UTC+1. Be aware that our application will be deployed on Openshift and is supposed to run without any manual interaction.

If the program fails to run or deliver data we want to be notified by email to the following e-mail addresses:

alexanderrnielsen@hotmail.dk

mikkelvig@hotmail.com

kiriian@hotmail.com

Vi skrev til alle 7 og bad dem besvare følgende spørgsmål:

Thank you for your reply, we are considering hiring you for the job, however before we can hire you we need to know how you understand the assignment, so we like you to answer some questions:

1. Could you explain how you understand the assignment? What do you expect to deliver? We ask this because some of the offers have a very large timeframe while others have a very small one, and we need to know if we have been unclear in our explanation of the product or if there is another explanation.

2. How much time in total do you estimate using on the project. How many hours in total? When do you expect to start and when do you expect to deliver?

3. Have you ever worked with scrapping websites before and how many times?

4. How experienced are you in working with threads in Java and which interface do you intend to use?