

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen

Group:

I det her Semester projekt havde vi mulighed for, at være 4 personer i gruppen, så på den måde havde vi allerede flere penge at bruge end nogle af de andre grupper som kun var 3. Da vi er 4 personer i gruppen havde vi 240 dollars til rådighed (60 dollars pr person). Vi har ikke brugt alle de penge vi har til rådighed, da vi fandt en at outsouce, som ville lave den del vi havde tænkt os at outsouce til 200 dollars, så i princippet sparede vi 40 dollars, da vi var klar til, at bruge alle de penge vi havde til rådighed.

Qualifications:

Det kan være rigtigt svært, at finde den rette person til det rette job, det fik vi lært first hand. For at finde den rette person til projektet var der mange faktorer, som vi kiggede efter, det var ikke bare hans eller hendes kvalifikationer, det er en person der er motiviveret og har evnerne til at opfylde de krav vi havde. For at sikre vi fandt den rette person, lavede vi en slags test, for at sikre de havde læst vores opgave, vi havde et keyword inde i vores dokument, som outsoucerne skulle skrive til os på den første besked, på den måde udelukkede useriøse bud fx folk der bruger roboter til, at byde for dem uden at have læst vores dokument. Vi fik afvide i klasse undervisningen, at der ville være mange forskellige typer af outsoucer, at enten ville prisen være rigtig høj eller rigtig lav. Da vi første lagde vores projekt op på freelancer.com fik vi mange forskellige tilbud, vi kiggede på deres kvalifikationer og stillede spørgsmål til nogle af dem. Vi fandt dog hurtigt ud af, at freelancer.com var useriøs grundet af de mange useriøse bud. Det var lidt som om de bare ville have jobbet uden, at overhovedet havde læst det vi havde skrevet, men bare hurtigt skimmet vores dokument. I vores Job beskrivelse krævede vi en person der kunne bruge ”

Skills Required

Java, Software Development, Database Programming, RESTful, Internet Security

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen

Efter noget tid gik vi videre til upwork.com, vi havde hørt fra nogle af de andre grupper at den var mere seriøs end Freelancer.com. Vi fik et rigtigt godt indtryk af upwork.com, da vi syntes hjemmesiden var stillet meget bedre op end freelancer, også de bud vi fik, havde alle sammen skrevet vores keyword i deres første besked. Der var dog igen lidt forskellige bud enten ville de lave det rigtigt billigt til 20 dollars ellers lå den til de 200 dollars. Så vi havde lidt den tankegang, at enten var opgaven nem at lave, så derfor ville de lave den til så lidt, også dem som havde budt højt bare ville tjene penge og havde overprised projektet. Så vi stod overfor for et valg, det var bare hvilken tilgang vi ville tage til projektet. Vi valgte så at gå med en person fra Croatien, personen havde fået rigtige gode ratings og havde 98% Job Succes. Grunden til at vi valgte ham, var fordi han stillede mange relevante spørgsmål til os om projektet, han havde forstået projektet og spurgte os om ting, som vi ikke havde tænkt over selv, det virkede som om han havde et rigtigt godt indblik til projektet.

2. User CRUD functionality (130\$)

- a. Database scheme, data,
- b. Web service API stubs,
- c. Database access



Steffen Lefort

allright deal

11:08 AM



Goran Molnar

OK. Gotta run now, we can schedule the meeting on Friday.
Nice to be working with you 😊

11:09 AM

Vi aftalte med ham han ville gå igang fredag den 29-04-2016 også skulle han senest være færdig tirsdag den 03/05-2016. Det var udfra de datoer vi også satte vores milestones.

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen


View Offer

PENDING - Expires on May 3, 2016.

Contract Title	Backend server with database and RESTful API		
Related Job Opening	Backend server with database and RESTful API		
Budget	\$200.00		
Milestone(s)	Name	Amount	Due Date
	1. Flight database &...	\$70.00 (funded)	Apr 29, 2016
	2. User CRUD funct...	\$130.00	May 3, 2016
Offer Date	April 26, 2016		
Client Message	<div></div>		
<div><div>Withdraw Offer</div><div>Make a Different Offer</div></div>			


Parties

Offer made by:



Steffen Lefort
Steffen Lefort

Offer made to:



Goran Molnar
Optimisation algorithms researcher

Related Links

[View original proposal](#)

[View messages](#)

TimeZone:

En af de største fordele og grund til, at vi valgte netop ham fra Croatien var nemlig fordi han også var i samme tidszone som os. Det var også en af de ting vi lagde meget vægt på, at vi skulle kunne snakke med personen under samme tidszone. Det kan nemlig være en stor ulempe at fx outsouce med en person fra indien hvor tidszonen ville være 5 timers forskel, så ville kommunikation godt kunne blive en ulempe, og det ville vi helst gerne undgå hvis vi kunne. Så kommunikationen var helt i top med os, vi kunne skype med personen og skrive med personen uden at det blev et problem.

Communication:

Kommunikationen med vores outsoucer har generelt været rigtig godt. Det var også en af de grunde til vi valgte ham, han snakkede flyende engelsk, også da vi skrev med ham via. Chat formulerede han sig rigtig godt. Vi har stillet spørgsmål og fået hurtigt svar og omvendt. Vi har givet ham vores skype, og han var ledig på skype, hvis der var spørgsmål eller forvirring benyttede vi os af skype, eller Upwork chatten. Vores outsoucer har pointeret nogle af de fejl som vi ikke selv havde tænkt

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen

over, og hvad der kunne være en bedre eller nemmere måde at gøre det på. I det hele taget har kommunikation været godt med ham.

Task:

Vores outsoucer havde til opgave, at lave vores backend del i vores opgave (the Metasearch Reservation System server) da vi følte det var det vi havde lavet hele semesteret, og følte os bedst ved, så ville det være sjovt at se hvordan en anden person ville lave den, og på den måde få indblik i hvordan en anden person ville lave det anderledes end hvordan vi har lært det. I vores kravsspecification havde vi beskrevet detaljeret hvad han skulle lave. Nedestående er det backend vi fik tilbage.

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen

Kravspekifikation

Disclaimer

This assignment is part of a larger school project that involves both backend and frontend technologies. As part of the assignment students are asked to outsource part of the work to be paid for and produced by freelancers. The final product is not a commercial product and will never be used in production.

The purpose is for our students to get experience with the process of writing requirement specifications and communicate these efficiently to developers. The project is not approved before all the agreed upon work has been paid for in full.

If you need to contact the academy please email either of the following addresses:

Lars Mortensen: lam@cphbusiness.dk
Thomas Hartmann: tha@cphbusiness.dk
Caroline Simonsen: chu@cphbusiness.dk

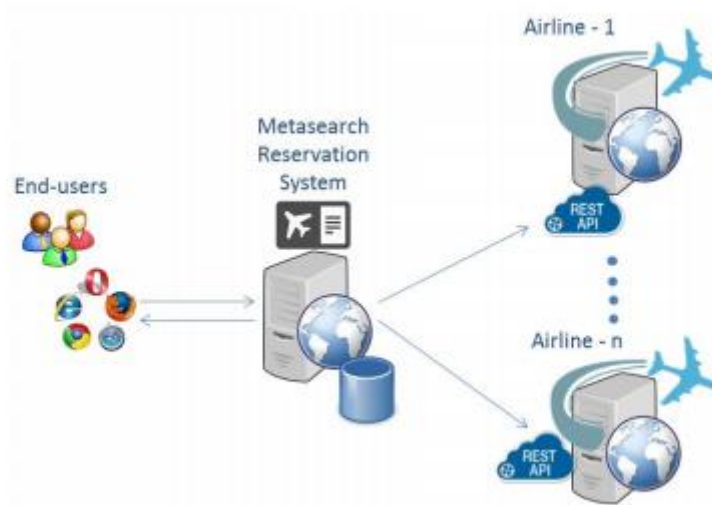


Description of project:

The system you need to make is the backend for the website of a fictitious company that deals in plane tickets. The full system is described in the diagram below (your job will be the Metasearch Reservation System server).

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen



The system will receive its data through several airline's (also fictitious) REST APIs. The system must then expose the data through its own REST api (including responses for erroneous requests).

The system also has to handle the creation, deletion and editing of both users and admins.

The system has to be able to receive request from the client on flights and seats, provided date, start location and destination, and amount of tickets.

Furthermore it has to save all requests/searches for tickets.

Please see the following for a dummy airline which must be used in development:

api/flightinfo/from/date/persons Example: *http://angularairline-*

plaul.rhcloud.com/api/flightinfo/CPH/2016-01-04T23:00:00.000Z/3

api/flightinfo/from/to/date/persons Example: *http://angularairline-*

plaul.rhcloud.com/api/flightinfo/CPH/STN/2016-01-04T23:00:00.000Z/2

Specific technical requirements which has to be met before project will be accepted as completed:

- ⑩ The system must be written in Java, for JDK 1.7 (the host we'll use for deployment doesn't support JDK 1.8!).
- ⑩ The system must use JPA ORM for all database operations.
- ⑩ All implemented database CRUD operations must be unit tested (preferably using JUnit).
- ⑩ The REST api too, must be tested either using unit tests or POSTMAN.
- ⑩ The password of Users and Admins **MUST** be stored in a hashed and salted fashion.
- ⑩ All dates and times **MUST** adhere to the ISO-8601 standard.
- ⑩ Database has to be MySQL.

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen

Requirements which has to be met before project will be accepted as completed:

- ⑩ Junit tests which document the creation and persistence of users and admins to a MySQL database.
- ⑩ POSTMAN (or similar) pictures which document a working (full CRUD) REST API which follows the format provided in this document.
- ⑩ Comments on all methods which describes what each method does and how.

Description of the REST API, JSON format and error responses

Information on the 2 get requests which the server will receive from the client.

How the JSON has to be formatted and how the error responses JSON has to be formatted

Method	URL	Description	Request-Data (JSON provided with the request)	Response-data (JSON response from the request)	Error
GET	api/flights/:from/:date/:tickets <u>Parameters:</u> from: Start Airport (as an IATA Code) date: Travel date (as a ISO-8601 date) tickets : Requested amount of tickets (integer)	Request all available flights that matches the provided search criteria's	None, besides the parameters given in the URL	See response-1 below	See FlightException below
GET	api/flights/:from/:to/date/:tickets <u>Parameters:</u> from: Start Airport (as an IATA Code) to: Destination Airport (as an IATA Code) date: Travel date (as a ISO-8601 date) tickets : Requested amount of tickets (integer)	Request all available flights that matches the provided search criteria			See FlightException below
POST	api/reservation/:flightId flightId: id returned by on the two GET requests	Make a reservation for flight and persons provided with request	See reservationRequest below.	See reservationResponse below	See FlightException below

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen

keyword: danishDynamite

JSON Format for Data Exchange between Server and Client

ID	JSON – Object
<i>Response-1</i>	<pre>{ "airline":String, "flights":[{ "date": ISO-8601 String (date+time), "numberOfSeats": Integer, "totalPrice": Number (Euro), "flightID": Integer, "travelTime": Integer (minutes), "flightNumber" : String "destination": IATA-Code (String), "origin": "IATA-Code (String) }] }</pre>
<i>reservationRequest</i>	<pre>{ "flightID": Integer, "numberOfSeats": Integer, "reserveeName": String, "reserveePhone": String, "reserveeEmail": String (valid email), "passengers":[{ "firstName":String, "lastName": String }] }</pre>
<i>reservationResponse</i>	<pre>{ "flightID": "Integer", "origin": "String (Friendly name + IATA)", "destination": "String (Friendly name + IATA)", "date": "ISO-8601-Date/time", "flightTime": "Integer (minutes)", "numberOfSeats": "Integer", "reserveeName": "String", "passengers":[{ "firstName": "String", "lastName": "String" }] }</pre>

JSON format for exchange of user info, between client and server:

JSON Format and HTTP codes for Errors Reported By the Server

All errors responses must be reported using both one of the following HTTP-Status Codes and a supplementary JSON Error Response given below:

HTTP Error Code

- HTTP-400 (all errors relating to the client side, like illegal data etc.)
- HTTP-404 (resource does not exist)
- HTTP-500 (For all Unchecked Exceptions Thrown on the Server)

FlightException - Error Response JSON

```
{
  "httpError": integer (400, 404, 500),
  "errorCode": integer (see below),
  "message": A descriptive message, meant for end users
}
```

How to interpret the *errorCode*:

- 1: No Flights
- 2: None or not enough available tickets
- 3: Illegal Input
- 4: Unknown error
- 10 -> You can define your own codes here

Outsourcing Experience

Kasper, Iman, Sebastian, Steffen

Method	URL	Decription	Request data	Response data	Error
POST	api/user/new	Informes the server that a new user or admin is to be made, using the data provided with the request.	See <i>user request</i> below		Same as <i>flightException</i> above, only without the "errorCode".
GET	api/user/:role	Returns all users in the database, with the given role.		See <i>all user response</i> below	Same as <i>flightException</i> above, only without the "errorCode".
DELETE	api/user/:userName	Deletes the given user/admin with the given username from that system.			Same as <i>flightExcpction</i> above, only without the "errorCode".

Name	JSON Object
All users response	<pre>{ "users" : [{ "userName" : String, "role" : String }] }</pre>
User request	<pre>{ "userName" : String, "passWord" : String, "role" : String }</pre>

Final notes: In case you decide to apply for this project, please include the keyword given in this document, in your initial message to us. Any applications without this part will automatically be disregarded.