

Otso Case CodeCamp - Winter 2015

Ilkka Tommola

Hatef Shamshiri

Michal Micor

Razaq Shonubi

Abstract

In modern business having a web page is not enough. To be successful it is required to gather data and information about customers, their needs and satisfaction. That's why programming is getting more and more required skill nowadays. It is applied in wide variety of different areas and one of main focus is gathering, processing, analysis and presenting data in desired form.

In this paper way to gather feedback from customers, analyze it and display results is presented. Outcome of our work is system collecting feedback from customers, storing data in database and displaying results.

1 Introduction to business case

Main idea of codecamp is to provide manufacturer app allowing to receive feedback from customers related to provided service and product quality. Performance of subcontractor is strongly affecting how customers perceive manufacturer, as subcontractors are direct provider of service to the customer. Therefore It is required to control subcontractor performance by collecting detailed feedback from customers.

In our approach PhoneGap mobile application prototyping software along with c9.io cloud computing services has been used to run PhoneGap server, database and create mobile apps. More detailed information related to technology involved in this project and the way it works is presented later in this paper.

Result is system collecting feedback from customers, storing data in database and providing display of results in user friendly form. It also allows manufacturer to inspect subcontractor performance and based on this information, decide if cooperation with particular subcontractor should be continued or terminated.

2 Program features and technical description

The goals for the project was to develop a mobile application or a web page for a manufacturer in order to make their customers able to give a feedback about manufacturer's products and services. Aside from creating an app with user-friendly interface for their customers, there is a platform where the manufacturer can manage their subcontractors activities. In addition, the manufacturer is able to choose a customer to sent a feedback request. Also on the manufacturer page, there is an option for viewing and managing assignments and products. Moreover there is also a platform where the subcontractor can see a "scorecard" of their performance and compare it to other subcontractors.

For every app separate, but similar structure has been created.

All three apps have the same login screen, by which one user (admin, subcontractor and customer) can login to the service.

1.1.1. Customer App

This type of account allows user to login to service as particular user, choose product to rate and provide feedback to manufacturer in three categories: product quality, service quality and delivery time. Feedback is based on rating from 1 to 5 but there is also possibility of leaving comment in each category. Moreover customer can choose, if he wants to receive answer for given feedback.

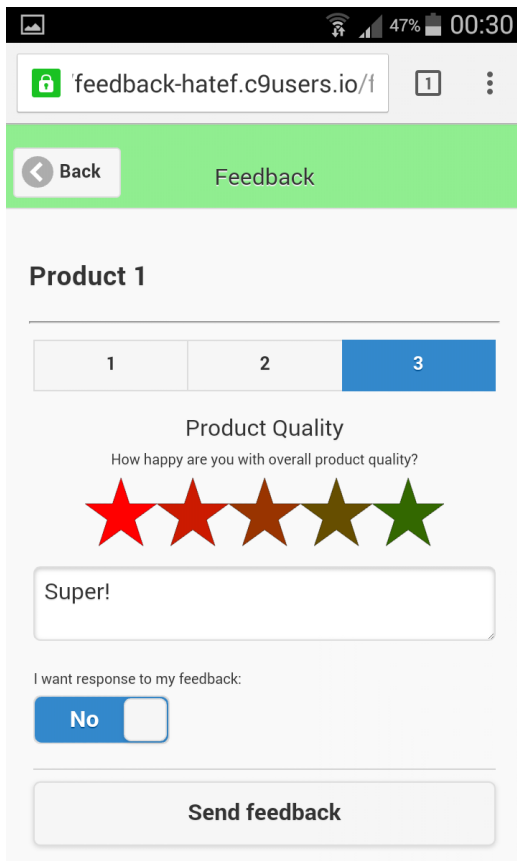


Figure 1 Rating input

At the end of process, after sending feedback, customer receives confirmation of sending feedback.

1.1.2. Subcontractor App

By this app subcontractor can compare his performance in reference to overall average performance of subcontractors. Performance is

presented in form of a bar graph. It is easy to read and compare subcontractor's personal rating with overall rating of other subcontractors. Clicking on graph's bar will show detailed information about rating represented in clicked bar.

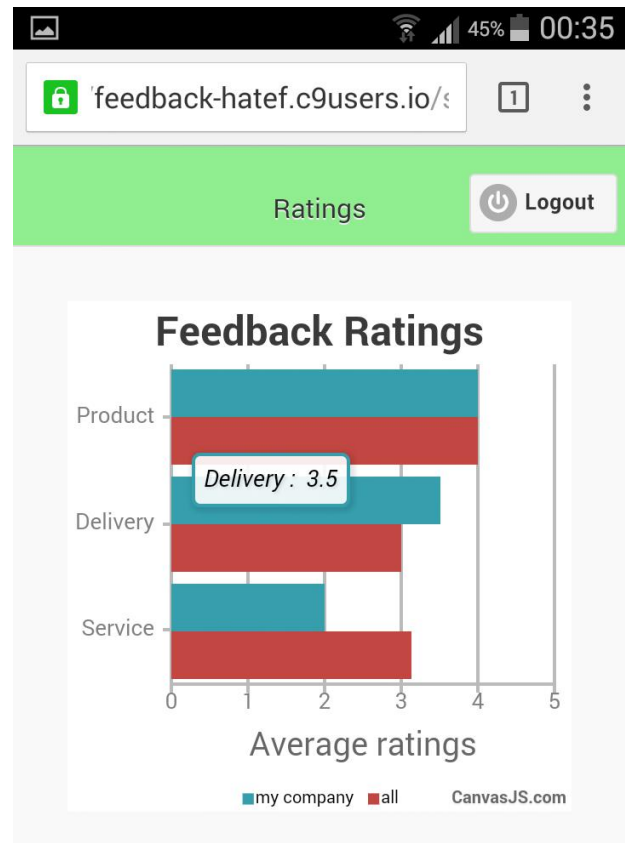


Figure 2 Feedback ratings

1.1.3. Manufacturer App

Manufacturer app allows for a variety of actions. Main focus is managing work of subcontractors, therefore manufacturer can assign and create orders and also add new product. Aside from that manufacturer can send feedback request to particular customers.

Actions are performed in step-by-step way. For example to assign order, user should click "Assign Order", from next menu choose product to assign and in next menu from drop-down list choose subcontractor.

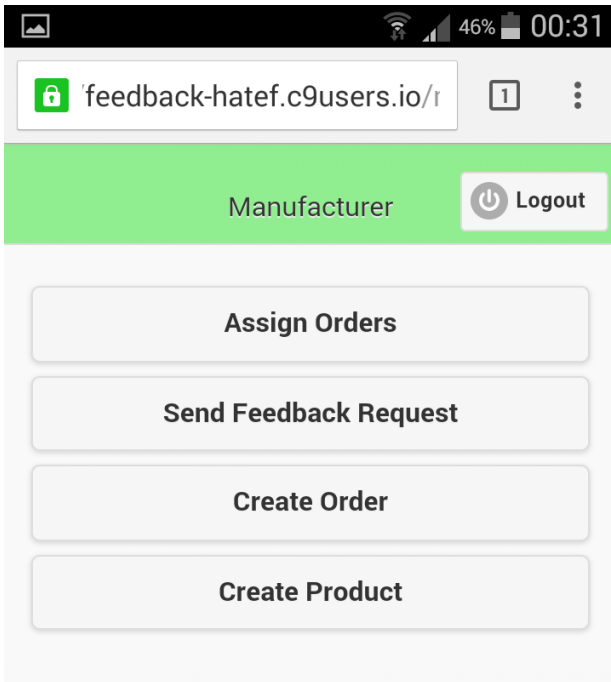


Figure 3 Manufacturer app main menu

Last step is to confirm chosen subcontractor by pressing “Assign” button.

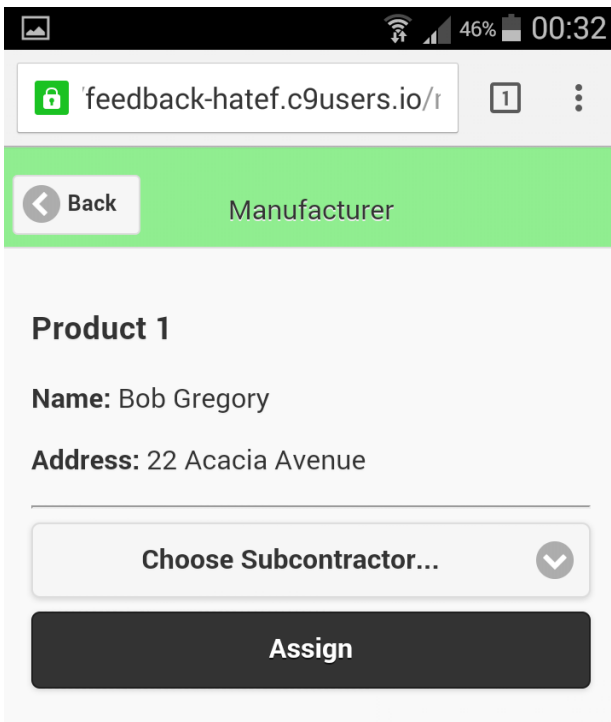


Figure 4 “Assign” window

3 Design

System is based on a web server which uses PHP and runs in cloud computing service c9.io. This server has a database in which all feedback, orders, subcontractors and all other additional information are stored. Furthermore 3 mobile apps prototypes has been created. Each one of them is running on PhoneGap server running also on c9.io cloud computing service. Their purpose is providing or displaying feedback and its results.

We chose not to use any fictional company as an example, and customize the whole application around the needs of one specific company. Instead we wanted to develop a very generic application that can be easily adopted by any company, and enhanced by adding company's own branding information like colors and logos.

3.1 Frontend

Frontend is part of our project responsible for collecting feedback data and displaying received data in proper form. Currently it is running on PhoneGap server (development tool for mobile app prototyping), but in future it can become separate app.

All content on mobile device is displayed in PhoneGap app which emulates native mobile application . Application which should be emulated can be downloaded from PhoneGap server running on c9.io cloud and executed on the PhoneGap Development app on the mobile device, or the service can be previewed on a normal web browser.

After app has been downloaded and started, user can login to admin, subcontractor or customer account. Then app connects to server containing database and performs actions requested by user like displaying subcontractor rating or sending customer’s feedback to database.

3.2 Backend

Backend part of our project is running on two cloud servers.

First server is responsible for logging in, storing and analyzing gathered feedback and data related to orders and subcontractors. It is written in PHP and all data is stored in PostgreSQL database.

When client app sends a request to backend, e.g. to list all orders, backend uses SQL to query the database and responses with a JSON object containing requested data. Or client application can input new data like orders or feedback, or update the data that is already in the database.

Second server is required to allow PhoneGap development tool to emulate our application. Both servers are running in c9.io cloud.

3.3 Technologies and libraries

As mentioned above frontend apps are working on mobile devices via PhoneGap application. We have chosen to use PhoneGap and JQuery mobile since using them allows fast prototyping. Additionally applications developed for PhoneGap can easily be deployed to variety of platforms that PhoneGap supports. Furthermore server required for apps to exchange data is written in PHP and it is equipped with PostgreSQL database. However the backend is designed to be very simple so that it can easily be ported to another platform if necessary. We chose to use PHP and PostgreSQL because using them was the simplest and fastest development option for us.

Full list of used technologies is given here:

- PhoneGap
- JavaScript
- JQuery Mobile
- JQuery
- PostgreSQL
- jRate

- CanvasJS
- PHP

4 Future work ideas

- Manufacturer and subcontractor apps should provide more detailed and sophisticated data. For instance info how well subcontractor is currently performing in relation to past periods of time.
- Manufacturer app might also provide basic aid in decision making process (suggestion about top/worse supplier performance or pinpointing rapid changes in performance time place in short amount of time).
- Manufacturer should be able to add/delete subcontractors.
- Subcontractor should be able to review individual feedback comments and respond to them via app if customer wishes to be responded.
- All user input should be sanitized both on client app and server. Currently the system is vulnerable to e.g. cross-site scripting attacks. If a user injects custom JavaScript code into feedback comments, this code will be run on the client app that is browsing feedback comments. On the course we skipped this issue partly because of time constraints and partly because it was mostly out of scope of this course.
- Manufacturer app (and subcontractor app, when it has more features) should have filtering options for finding the relevant information. For instance when there are hundreds of orders,

manufacturer should be able to select from recent orders or find customers by name, address etc.

- Communication between client and server should be optimized in some cases. When database has large amount of data, it is not feasible to send the full list of orders or products to client.
- All three apps should be made more responsive. Currently the problem with some application sub pages is that the page layout is fully rendered only after the app has received full response from backend. When response is greatly delayed, page will be rendered in two noticeable phases and the first one looks weird as style sheet (CSS) hasn't been applied yet.
- Adding/removing rating categories should be more flexible. As an example we have used three categories – service quality, delivery time and product quality – for ratings, one tab per rating. Manufacturers should be able to pick which ratings are available for feedback and customer app should generate tabs dynamically based on manufacturer's preference.

We didn't implement any genuine authentication method for our system as the companies using feedback app should already have some kind of authentication system available. Each system would need its own plugin in order to enable real authentication. Instead we used "client side authentication" to emulate login, and redirect users to appropriate pages.

5 Conclusions

To summarize our project, we can describe it as complete modular system, based on server with database and mobile application performing their roles. Apps can be modified and upgraded anytime, without affecting entire system.

Even as a prototype, apps are allowing to perform actions as normal app, making the whole system fully operational, despite it early development stage.

This feature gives huge advantage and benefit in developing systems and apps like these. Developers can introduce product to client and test if it suits customer needs by providing emulation of raw functionalities, even before they start working on app dedicated for particular mobile hardware. Moreover, any application created using this method can be quickly and easily deployed to any platform that is supported by PhoneGap.