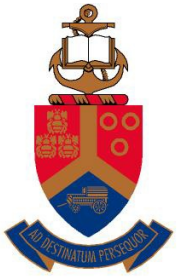

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Accompany
Requirements Specification and Design

18 April 2017

Nokuthula Manana 12064115

Contents

1 Project Background

Accompany is a simple company website that provides information about the company, its vision and its trade. This website is intended to be a static, informative site.

2 Vision and Scope

The core of the system will be a single page application that dynamically loads different content describing the company based on which tab is selected. The website will allow any user to be able to send a query to the company to find out more information.

3 Design Specifications

The system should be designed in such a way that the mailing service can easily be plugged in and out of the system i.e. it should be loosely coupled from the rest of the website. The following are the design specifications for the website:

- Any updates to the website will be handled by the administrator. The website prototype will be hosted on the UP server in the research labs for the duration of the research being conducted.
- The Anonymous User interface will allow people to browse the entire website
- The mail service that will enable users to enter their contact details should they want to be contacted by the company. The mail service should send an email or SMS notification to the relevant people within the company to handle queries.

4 Interface Designs

Below are the prototype designs for the website:

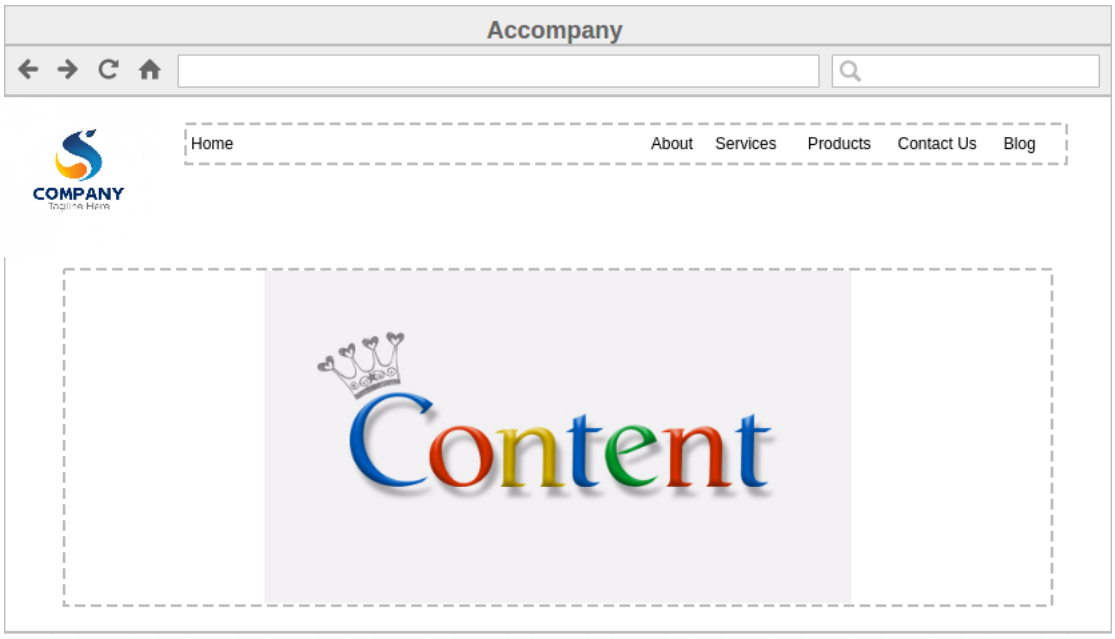

Figure 1: Admin Home Page

This is a single page application that will basically update the *content* component when a new navigation link is clicked.

Figure 2: Admin Home Page

This modal will pop up when the *Contact Us* link is clicked by the user.

5 Design Technologies

The following technologies will be used to implement the system:

- Html 5 (Html and Bootstrap CSS)
- Angular2
- Spring Boot
- Spring MVC
- Apache Maven
- Git (Github)
- PostgreSQL Database

6 Architectural Technologies

The web application will consist of two subsystems that communicate via HTTP using REST Framework. The Java/Spring Boot application will be known as the "backend" application. The HTML5/Angular2 application will be known as the "frontend" application. The backend application is expected to communicate with the database and use Maven as its dependency management tool whereas the frontend application will be hosted in the browser and NodeJS is expected to manage packages required for the application to run successfully.

The backend application will be implemented twice. The first implementation will be under a monolithic architecture while the second will be under a microservices architecture. The frontend application can be used in both the monolithic and microservices versions of the backend application. It will make calls to the frontend application by means of Angular 2's Http module that will target the REST API endpoints of the respective backend applications.

7 Change Factor

In order to fully test the abilities of microservices architecture a change factor must be introduced to the system once implementation has been completed. This change factor should be significant enough that it can demonstrate the differences of the architectures with this implementation.

The change factor that will be implemented is **user registration**.

A potential client/user should now be able to register on the system. Once registered there should be an interface provided for a registered user. This interface must include the following:

- A log of all the interactions with the company
- Information about their profile on the system

Additions to the user interface may be added if desired but these additions must serve a purpose.

8 Software Metrics

Below is a table of the software metrics of the current monolithic and microservices systems.