GMIT
Galway-Mayo Institute of Technology
B.Sc.(Hons) in Software Development

Project Best Ever Me Platform

Martin Repicky

Advised by Damien Costelo
Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Abstract

This project was initially set out to create a reverse auction cross-platform mobile application for Best Ever World ltd. This system's primary component was defined to provide a Client with platform to present their service requirements and allowed Service Providers to present their bids to this requirements. Secondary component was defined to provide a Service providers to advertise their Last-minute offers to Clients. Clients subscribed to categories of their choosing would be notified of these Last-minute offers and would be able to accept them.

The aim of this project was substantially modified to meet the company needs. The Best Ever World ltd is in the early stages of development and can be described as "Start up". This means that it has numerous active projects and future developments plans in different stages of development. This was discussed in numerous regular meetings with company CEO Portia Quinn, where company needs and plans were presented. This led to significant changes to project requirements to incorporate existing and future products under the one platform.

Unified platform was required to allow access all products and services with use of single login credentials. In order to develop this platform it was required to connect different frameworks together and allow them clear, structured and secure communication. Spring framework is used to create a server-side application based on Microservices architecture and client-side applications which include website and Ionic based mobile application. These are developed using Angular platform. The communication between all of these components is designed using Open API standards enhanced by Swagger toolkit. And security is maintained by the use of JSON Web Tokens. The combination of open source technologies was chosen to create a robust but modularized system where communication between all of the components are well documented and can be modified from single point. This will allow future development to be outsourced to different developers while keeping integrity of the whole system intact. The Best Ever Me ltd, doesn't have an in-house developers and is relaying on external developers to create and maintain their products and services, which makes this feature an important aspect of this project.

As a result this project main aim is to provide unified platform as was able to do so and manage to deliver a website and and Ionic mobile application for reverse auction and last-minute offer as proof of concept build on top of this platform.

Contents

Abstract:

Content:

                - Appointment
                - Messages
                - User Management
                - Metrics
                - Health
                - Configuration
                - Audits
                - Logs
                - API
                - Internationalization
        - Microservices
            - Request Service
            - Last Minute Service
        - Ionic Reverse Auction Mobile application

System design

- Database
    - Purpose
    - Schema
- Architecture decisions
    - Robustness
    - Modularization

Technology review

- Database
    - MariaDb
    - Ehcache
    - Elasticsearch
- Backend
    - Spring Framework
        - Spring Security
        - Spring MVC REST
        - Spring Data JPA
    - Jhipster Registry
- Frontend
    - Angular 5
    - Sass
    - Ionic
        - Phonegap
        - Cordova
- Developing Tools
    - JHipster
    - Gradle
    - Yarn
    - Docker

Conclusion

- Clients current and future needs
- Technology used
- Scale and Productivity
    - Complexity
    - Tools used to increase productivity

Introduction

Summary

Best Ever Me is aiming to become the platform where small and medium businesses focused, but not limited to Health, Beauty and Wellness and meet their online needs. Place where potential clients can research, find and contact service providers. Best Ever Me (BEM from now on) will provide the place where potential customers research their concerns about their health, beauty and wellness as a whole, find treatments and contact local or national providers for procedures of interest. BEM will also provide tools for service providers to operate their business online. These tools will include but no limit to the Booking system, online marketing, last minute service to fill cancellations and bidding on clients requirements in reverse auction system.

Github Repository Structure

Project main github repository is structured as follow

Gateway

This contains a Spring application which can act as monolithic application on it's own and is responsible for authentication and routing. Also contains a Angular based website under /src/main/webapp folder. Which include web interface for the rest of the microservices as well.

JHipster-registry-master

This is runtime application provided by JHipster team. It is an Open Source application. It consist of Eureka server, that serves as a discovery server for applications, as well as Spring Cloud Config server, that provide runtime configuration to all applications.

Request

This is a microservice application to provide reverse auction service for Clients to publish their service requests and Service providers to bid on them.

Lastminute

This is a microservice application to provide Last-minute offers from Service providers and Clients to subscribe to.

BemApp

This is a Ionic based cross-platform mobile application which is configured to interact via gateway application with Request and LastMinute microservices.

Choices

Best Ever World ltd is currently currently providing numerous services and actively developing new services in different stages of development and deployment.The main aim of this project has to incorporate all the aspect of the business and provide unified platform for these services.

Best Ever Me website (https://www.besteverme.ie)

The website is currently up and running It provides a place where users can research their health and beauty concerns and locate treatment providers nationwide. Service providers can register for free and advertise services.

Absolute Beauty Online website

(https://www.absolutebeautyonline.com/)

The website is currently under construction, but already online running in Beta mode. This website focuses on providing an online marketing service for small and medium businesses focusing, but not limited to Facebook ads platform. Future goals are to create and maintain clients online portfolio, that will include clients own website, Facebook and Google ads campaigns.

BEM reverse auction and Last minute offers

The service will provide an access via a cross-platform mobile application (Ios and Android) as well Angular base interactive website to a reverse auction service and Last minute offer service.
Reverse auction service will provide users with space they will advertise their service needs and service providers will bid on those requests. The user will then decide which offer it will accept based on price, location and feedback from previous users. The service will provide a means to pay for these services via Stripe payment system.
Last Minute service will provide a space where service providers can advertise discounted services due to an unexpected opening slot on their schedule. These services will be categorized into different categories which potential clients can subscribe to. This will allow them to receive notifications if service of their interest becomes available.

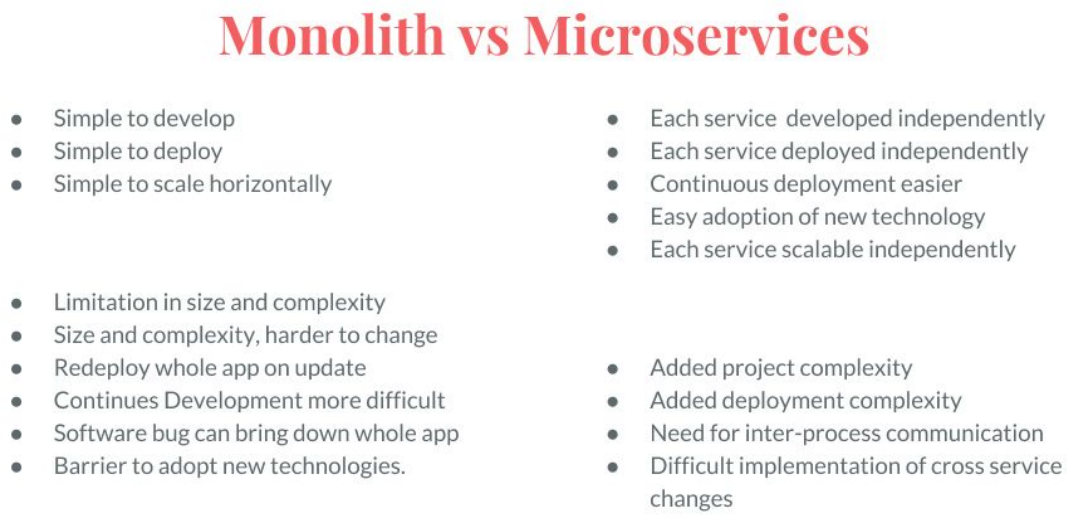
Unified BEM platform

The platform is aiming to unite current and future services provided by BEM. This will allow users, both Service provides and their clients to log in to a centralized hub with single credentials and use all the services listed above. The platform is developed in Spring boot framework backend with JWT authorization and Angular based front-end. The emphasis was put on modularization and potential scalability. This centralized platform will enable to better

analyze users habits and needs with strong user privacy policies in mind. This will allow BEM to better understand the market for health, beauty and wellness and provide paid consultancy for businesses to better adapt to current trends and market needs in general.

Platform Architecture

Due to the facts listed above a decision was made to create this project using Microservices architecture. Following criterias were considered in making this decision.



Source Control

The need for logical way to organize and control source code development is crucial part in modern day software development. This allow for better development by multiple developers and keep different versions of source code centralized. Source code is kept in Graph like structure and previous versions of code can be downloaded, forked and further developed. Development of this project was utilized on GitHub website and can be accessed by link bellow.
https://github.com/MartinRep/BEMmicro

Context

This project consist of

- Authentication system based on JSON Web Tokens for secured communication
- Management and System Administration web application
- Microservices for reverse auction and Last-minute offers
- Ionic based cross-platform mobile application

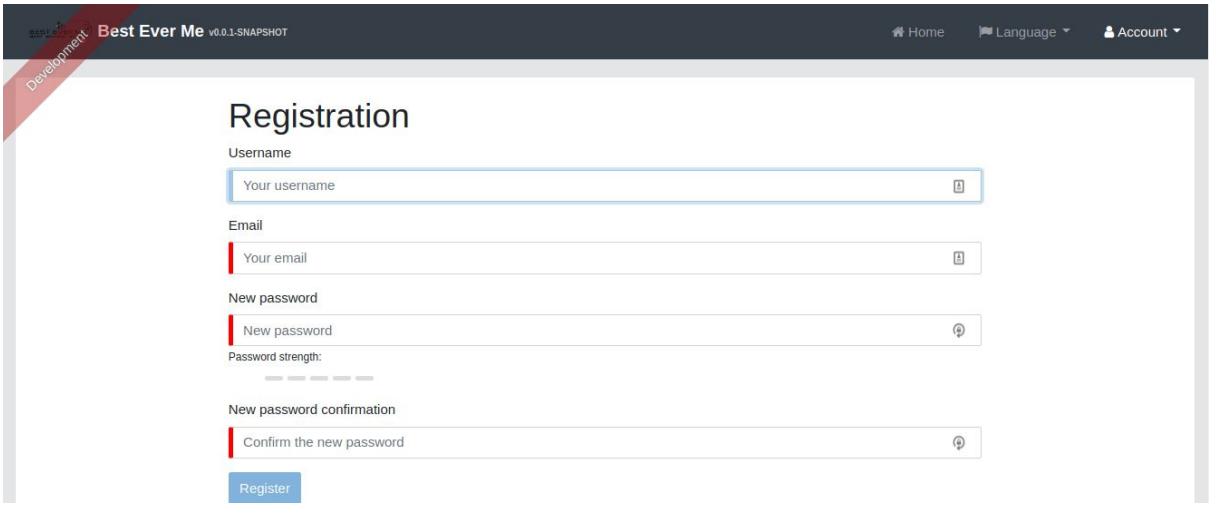
All this components of this project will be detailed below.

Authentication and Authorization System

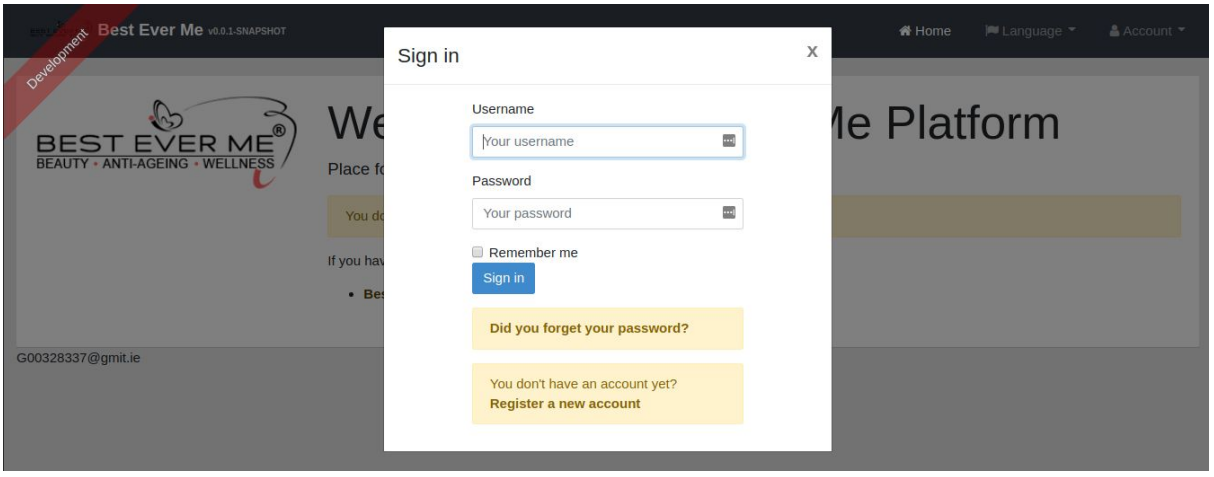
- Authentication is the mechanism whereby systems may securely identify their users. Verification of user identity via login and email confirmation.
- Authorization is the mechanism by which a system determines what level a particular authenticated user should have to secured resources. Permissions the user have to services via JSON Web Tokens.

Authentication

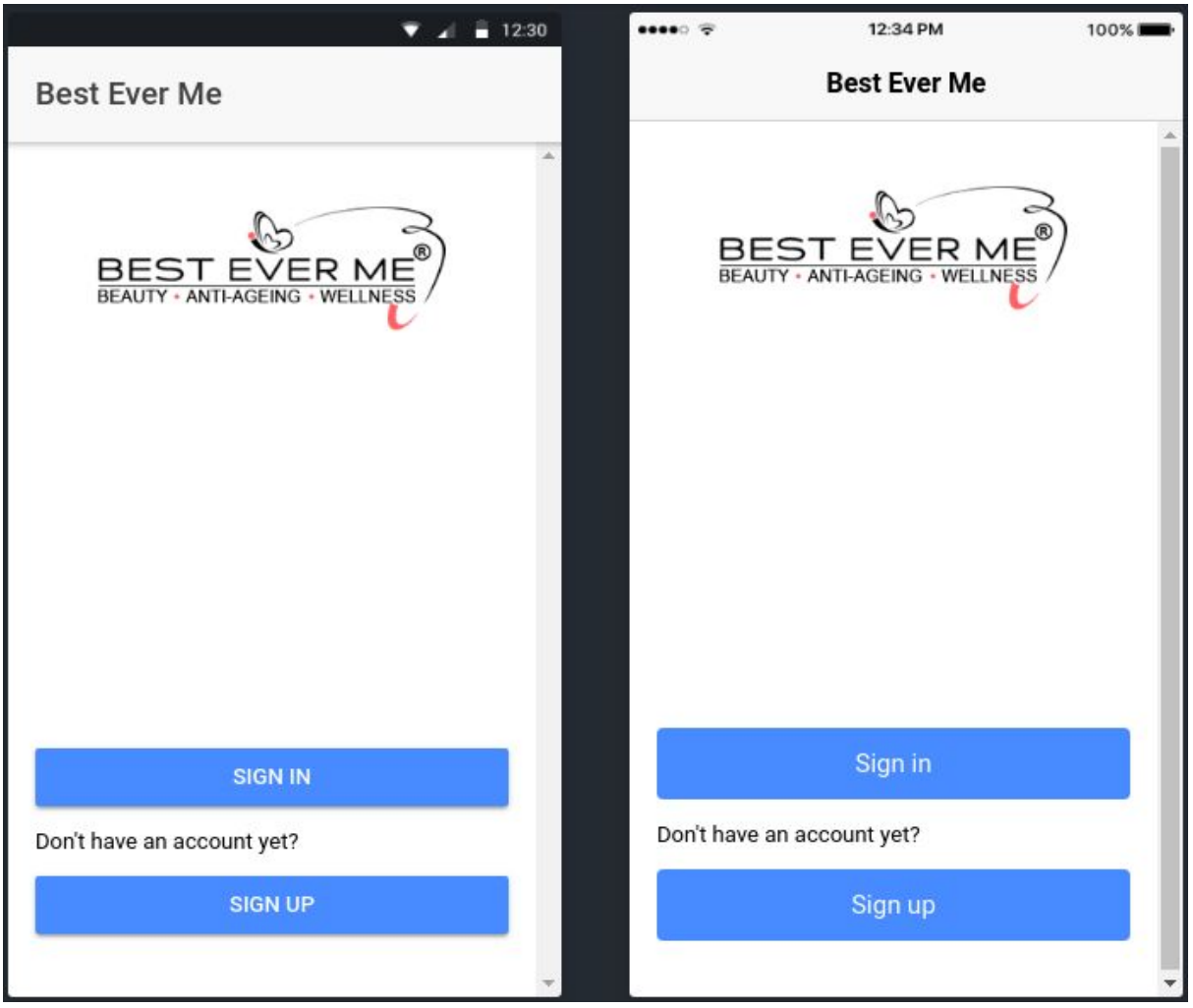
Gateway Web Interface
User authentication on client side in web interface is designed to authenticate user first by registering with unique username. Here user create username/password login credentials. Username is a unique and cannot be changed. Password can be changed, but has to comply with minimum requirements.

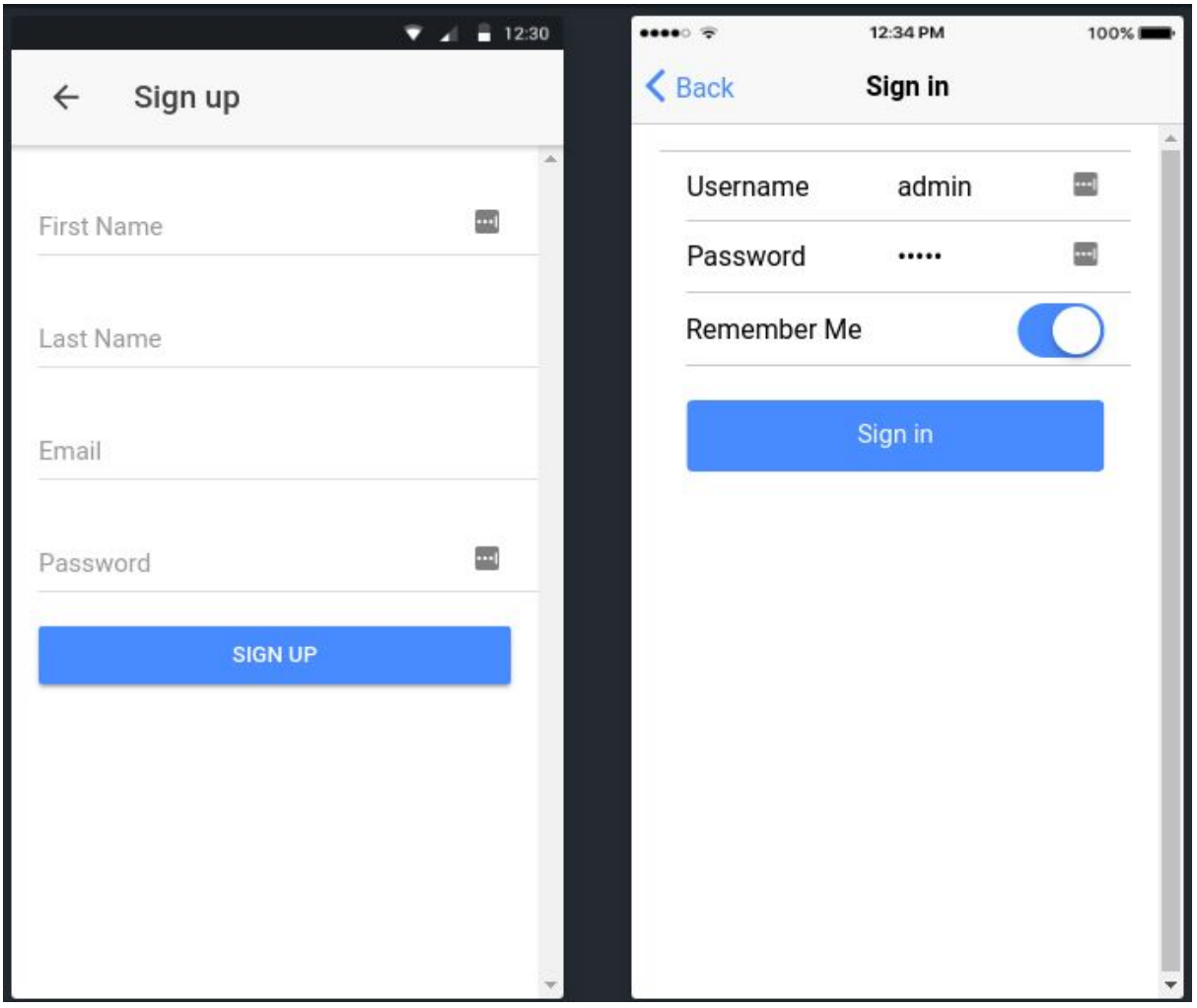


These credentials are than used to access all services and products in whole BEM platform ecosystem.



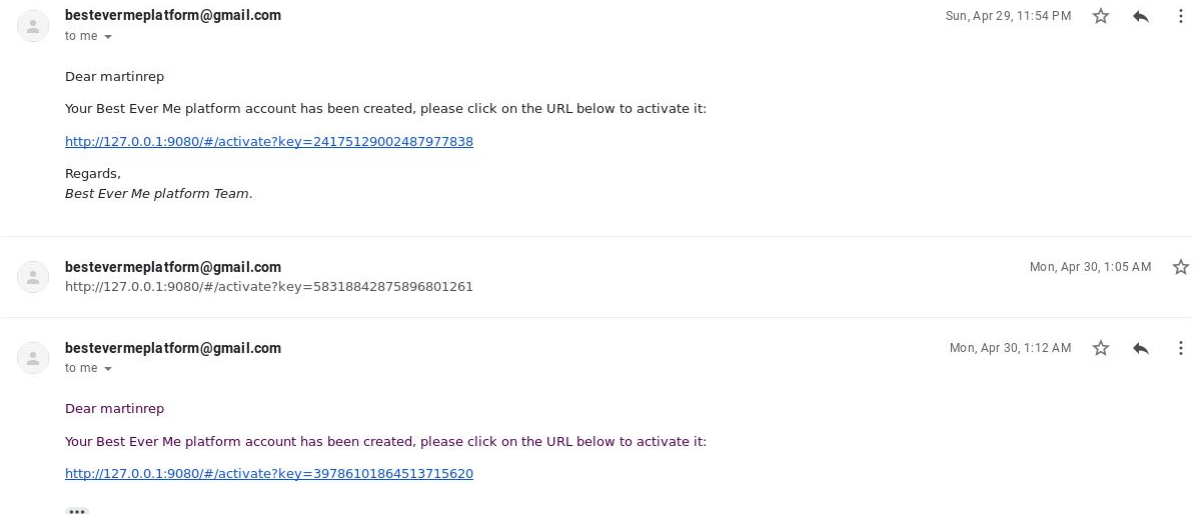
Simillary Registration and Login functionality in Ionic environment can be used to create users unique credentials and then can be used to access the reverse auction mobile application prototype. This is illustrated below in both in both Android and IOS environments.





Email Verification

This authentication feature is important part of users identity verification. After the user register and before they can log in for the first time, they need to confirm their email address by clicking on unique, auto generated link to confirm their identity. Email verification also serves additional function of creating a link between user and their email for the case of retrieving login functionality. This is done by did you forgot your password link in log in section. This will enable user to reset their password and enable them to renew their access to BEM platform.



Authorization

JSON Web Tokens
JWT (JSON Web Token) is an industry standard, easy-to-use method for securing applications in a microservices architecture.
Tokens are generated by the gateway, and sent to the underlying microservices: as they share a common secret key, microservices are able to validate the token, and authenticate users using that token.
Those tokens are self-sufficient: they have both authentication and authorization information, so microservices do not need to query a database or an external system. This is important in order to ensure a scalable architecture.
For security to work, a JWT secret token must be shared between all applications.
To share this key between all your applications, copy the key from your gateway to all the microservices, or share it using the JHipster Registry's Spring Config

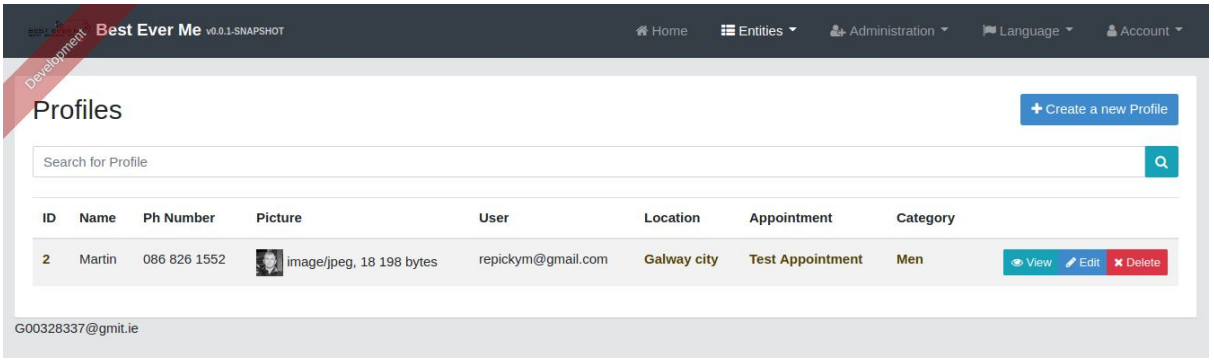
Management and System Administration Interface

Gateway Web Application

The web interface is created by open source Angular framework which will be described more in detail in technology review section.
In this section different functionalities will be described.

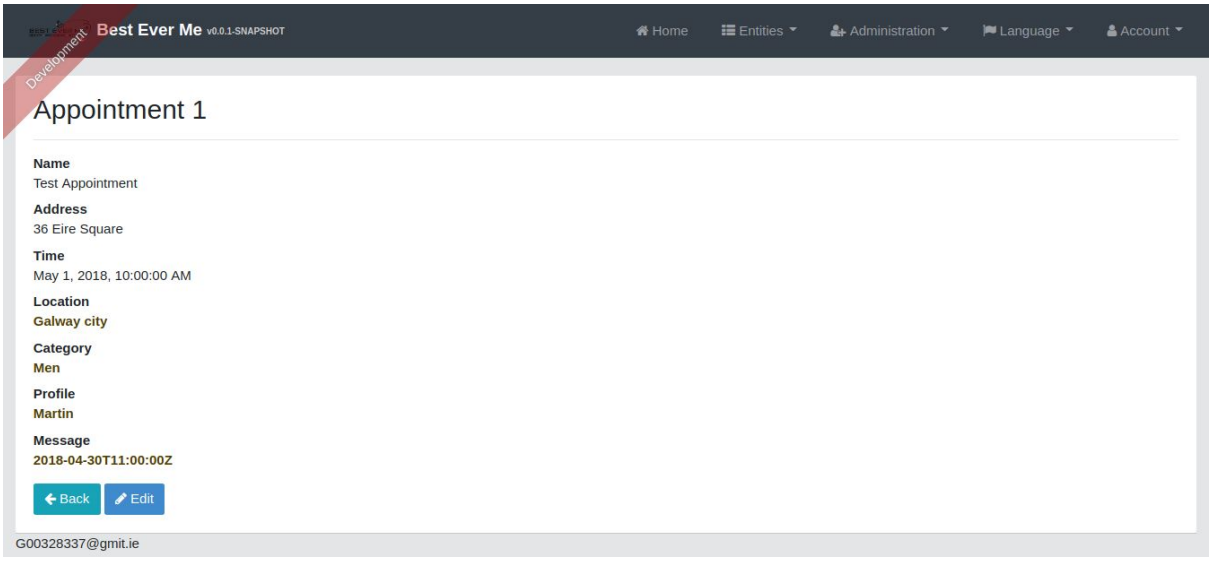
Profiles

Any user can multiple profiles. This provide a single user to act in behalf of multiple Service providers, business outlets and in the same time they can use the platform for their personal needs as client. This functionality was designed with small businesses in mind, such as self-employed hairdresser.



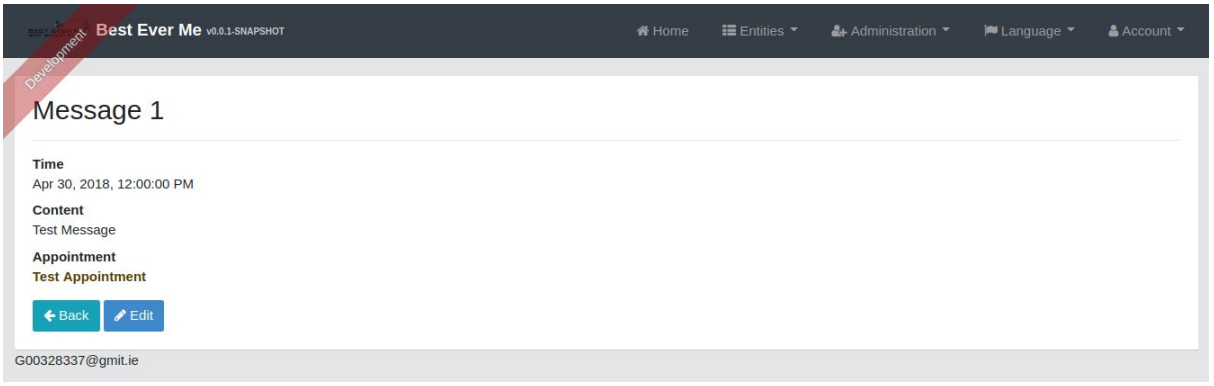
Appointment

Another core functionality common for all the services is appointment and that's the reason it was incorporated into Gateway application. This allow all the microservices to share appointments. Appointment is created when a bid for reverse auction request is accepted or when Lastminute service is accepted by client. This appointment is available to both client and service provider profiles only.



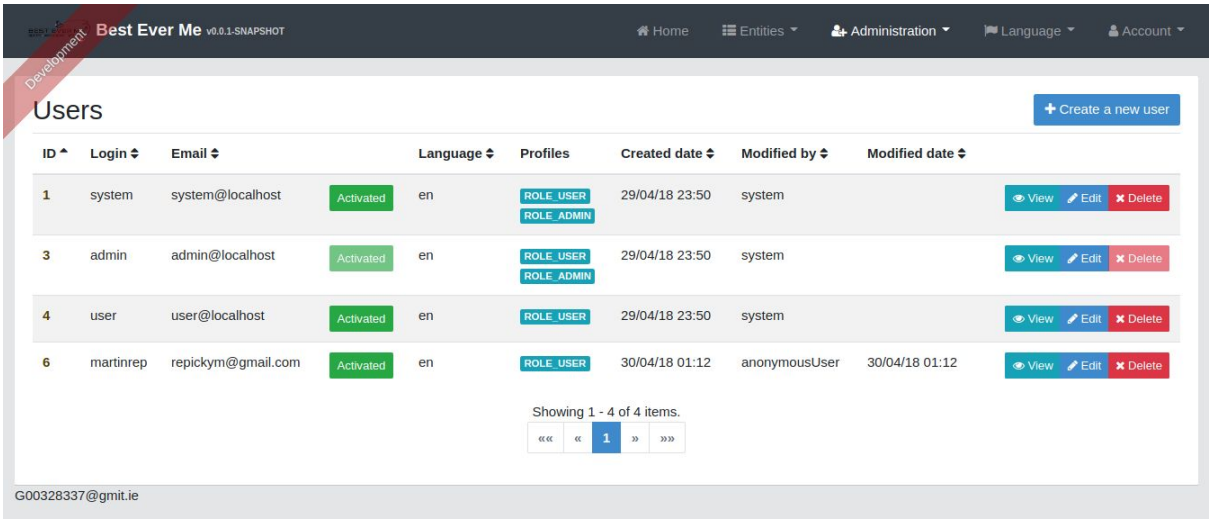
Messages

Message functionality is attached to an appointment and messages are accessible only by participants of the appointment. The purpose of this is to keep both parties in touch in case of cancellation and rescheduling. The choice was made to make this communication available through BEM platform to keep customers using the platform and to keep the track of such a communication in case of disputes.



User Management

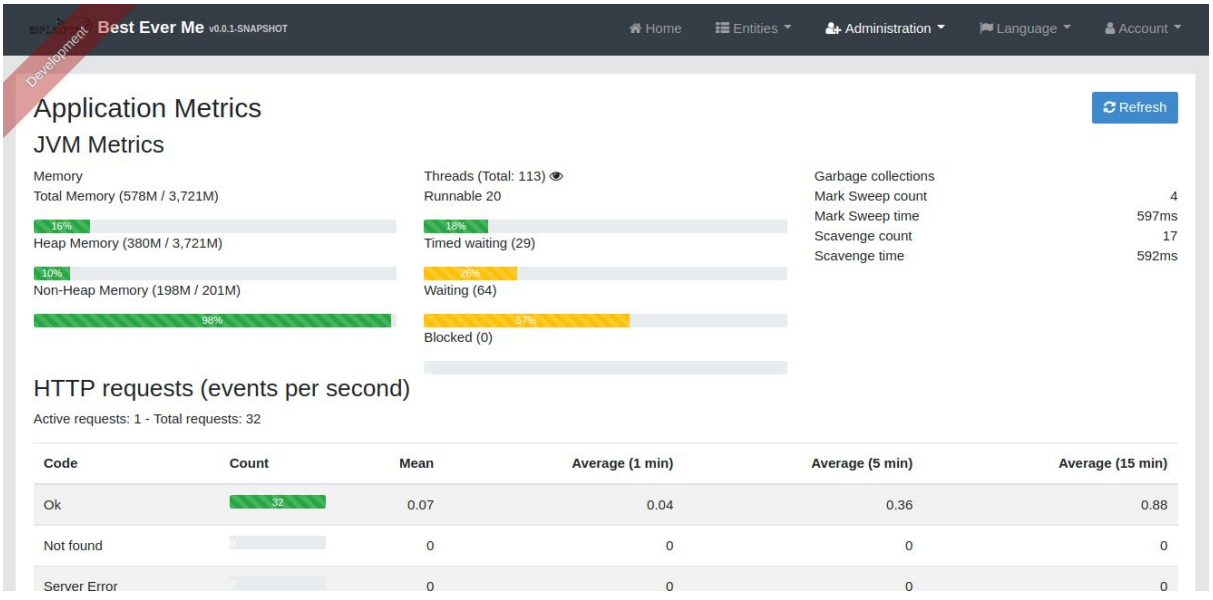
User management is an administrative tool to manage Users of the platform. Here administrators of the platform can view, activate, even edit certain fields, and delete users profiles.



Metrics

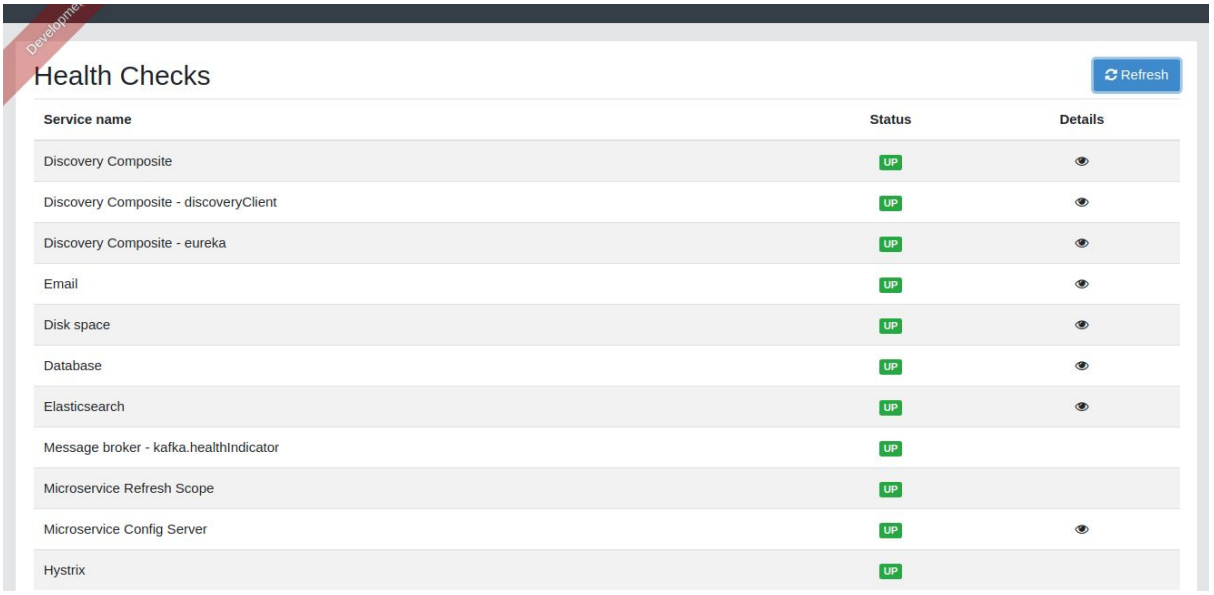
This administrative function servers to monitor underlying system resources on which platform is running. This is provided by metrics from Java Virtual Machine.
The metrics dashboard uses Dropwizard metrics to give a detailed view of the application performance.
It gives metrics on:

- JVM
- HTTP requests
- methods used in Spring Beans (using the @Timed annotation)
- database connection pool



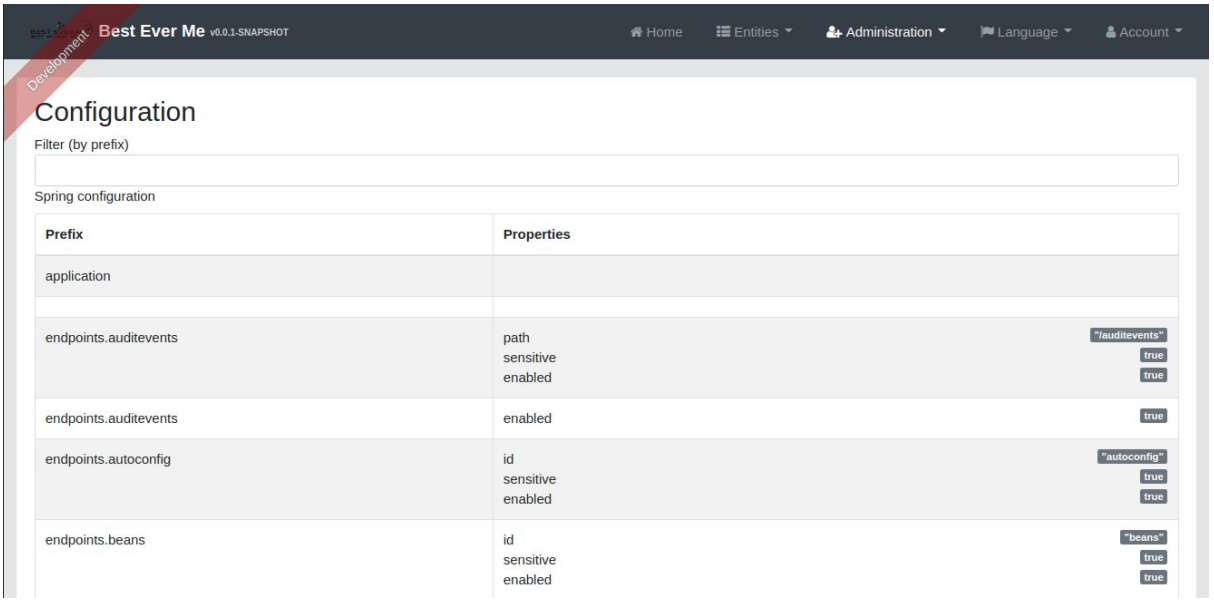
Health

The health dashboard uses Spring Boot Actuator's health endpoint to give health information on various parts of the application. Many health checks are provided out-of-the-box by Spring Boot Actuator, and it's also very easy to add application-specific health checks.



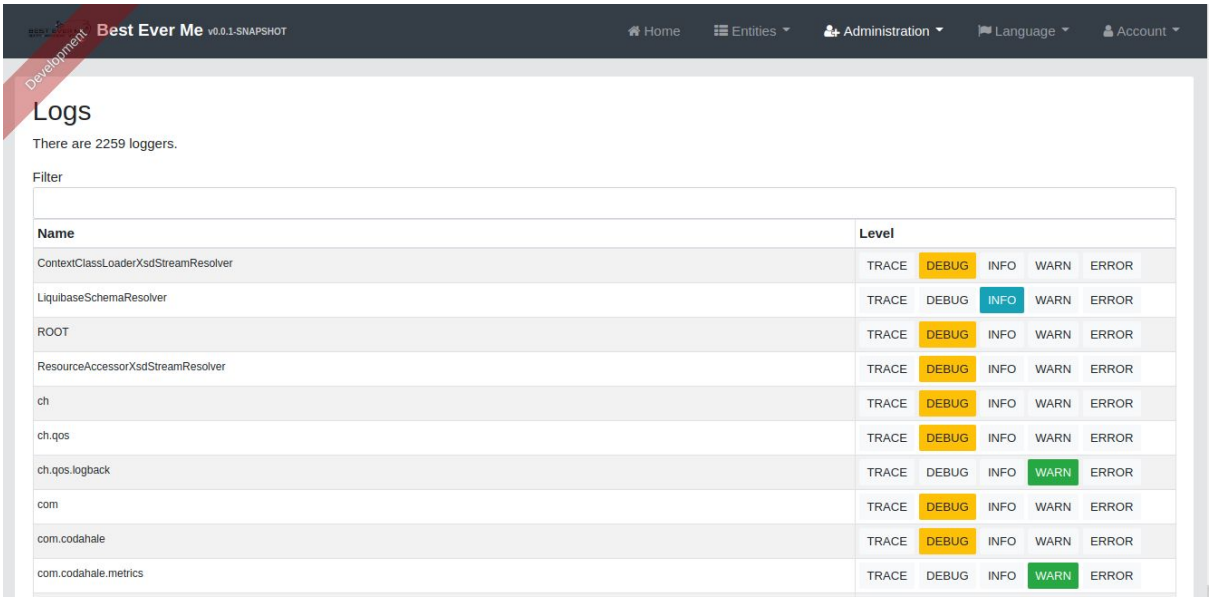
Configuration

The configuration dashboard uses Spring Boot Actuator's configuration endpoint to give a full view of the Spring configuration of the current application.



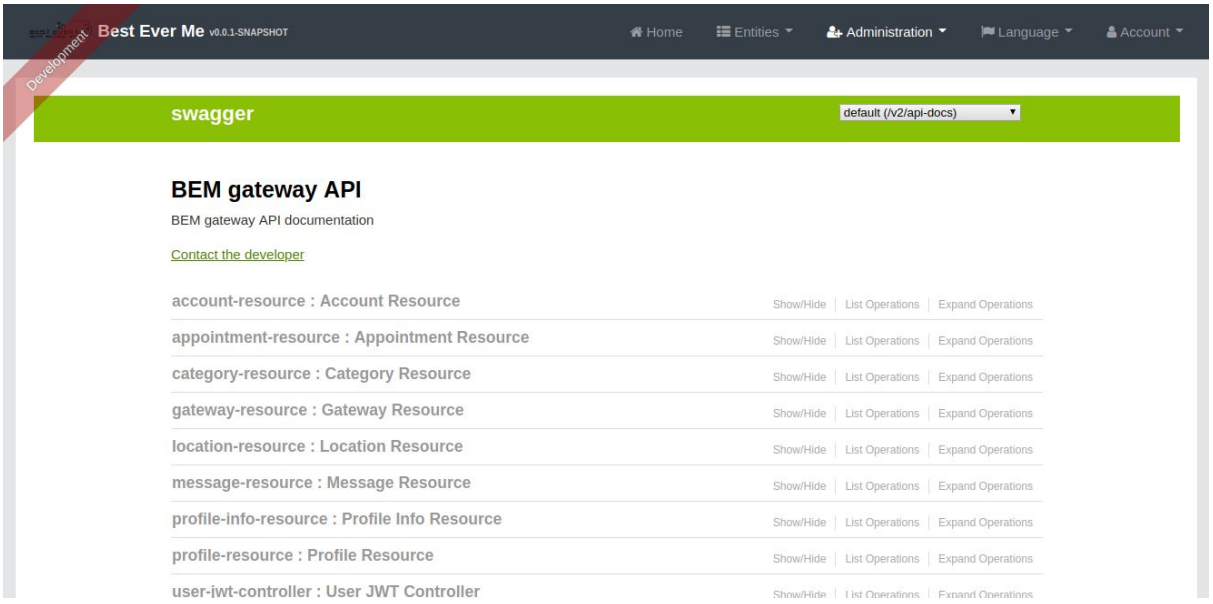
Logs

The logs dashboard allows to manage at runtime the Logback configuration of the running application. Changing the log level of a Java package is as simple as clicking on a button, which is very convenient both in development and in production.

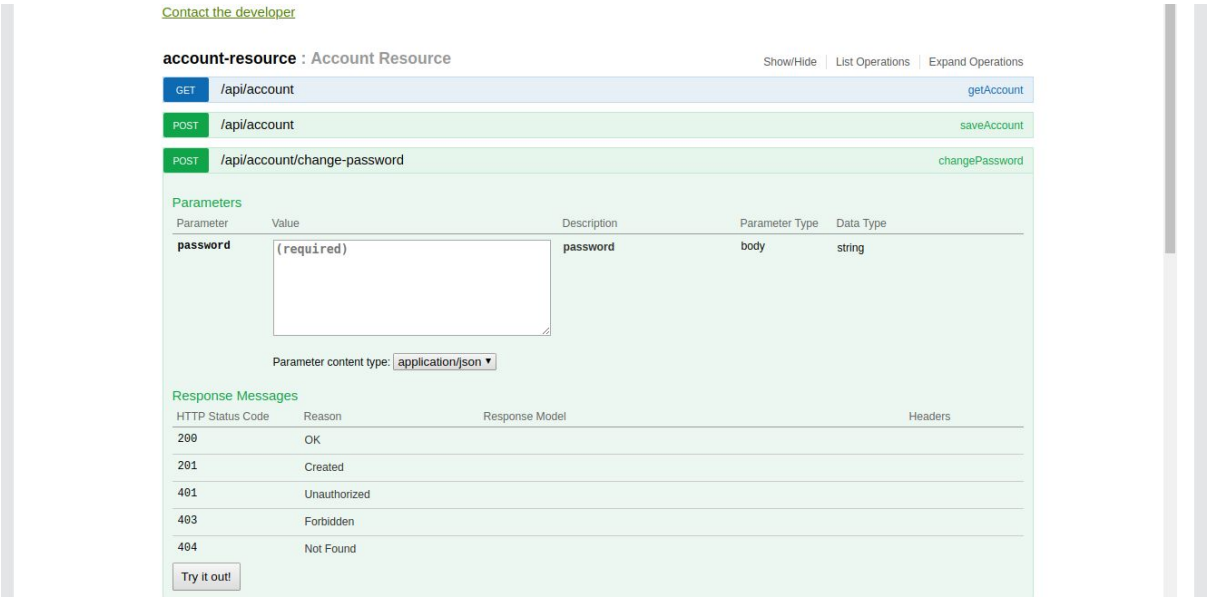


API Dashboard

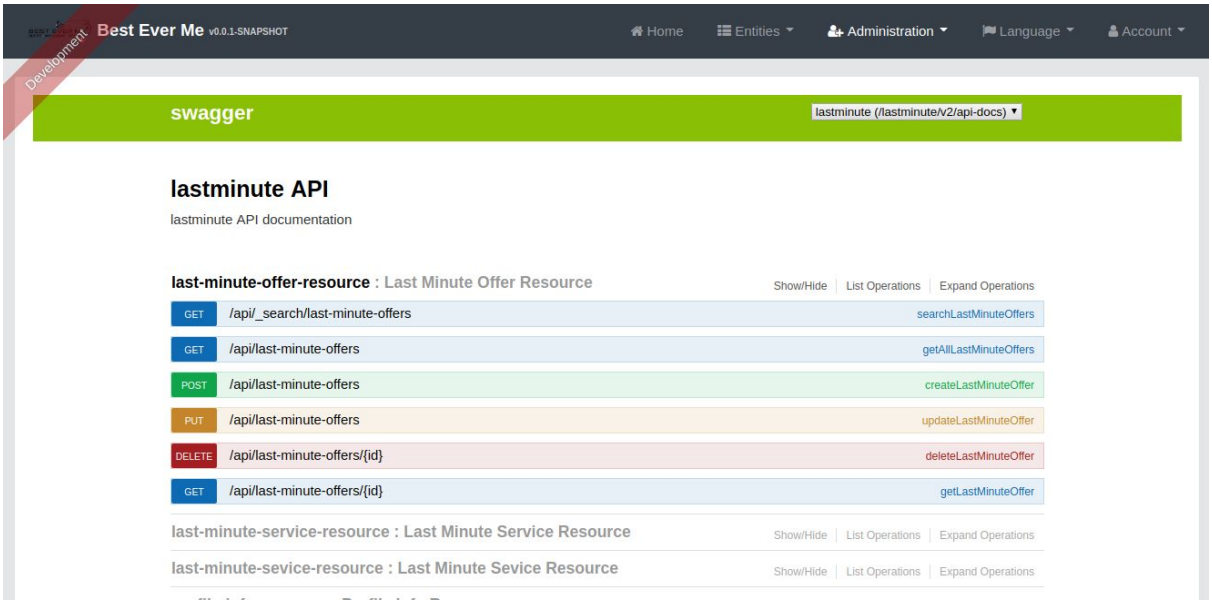
This exposes one of the core functionalities of BEM platform. Using swagger libraries to better describe and test every aspect of an Open API design. This include API of an gateway itself and all the microservices routed through gateway as well.



For every API endpoint there is a inbuilt test option functionality that will produce accurate results from response.

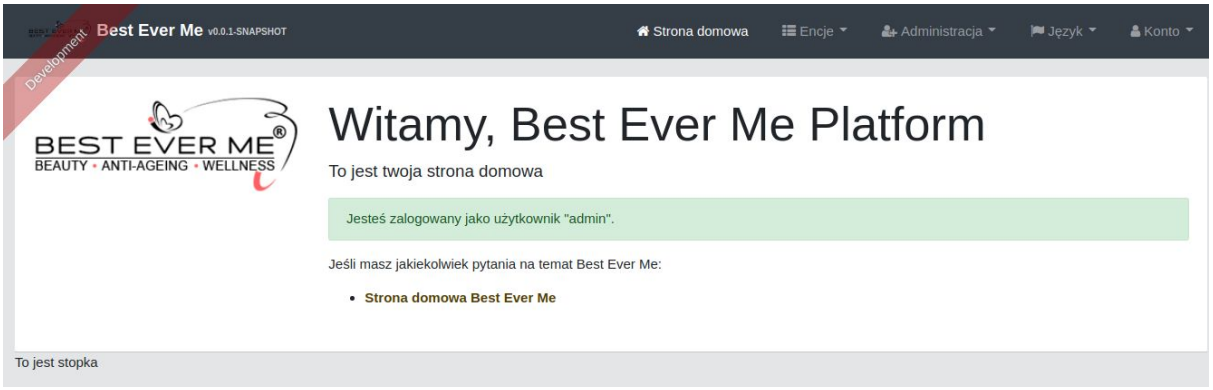


Picture below describe an actual miscroservice of Lastminute offers and its API endpoints. This is a very useful tool for future development.



Internationalization

Core functions of BEM platform are using optimized for internalization. Currently there are two additional languages implemented on top of the default English language and those are Polish and Slovak languages. Additional languages can be easily added in the future just by JSON file. This files are located under numeronyms i18n (where 18 stands for the number of letters between the first i and the last n in the word internationalization).

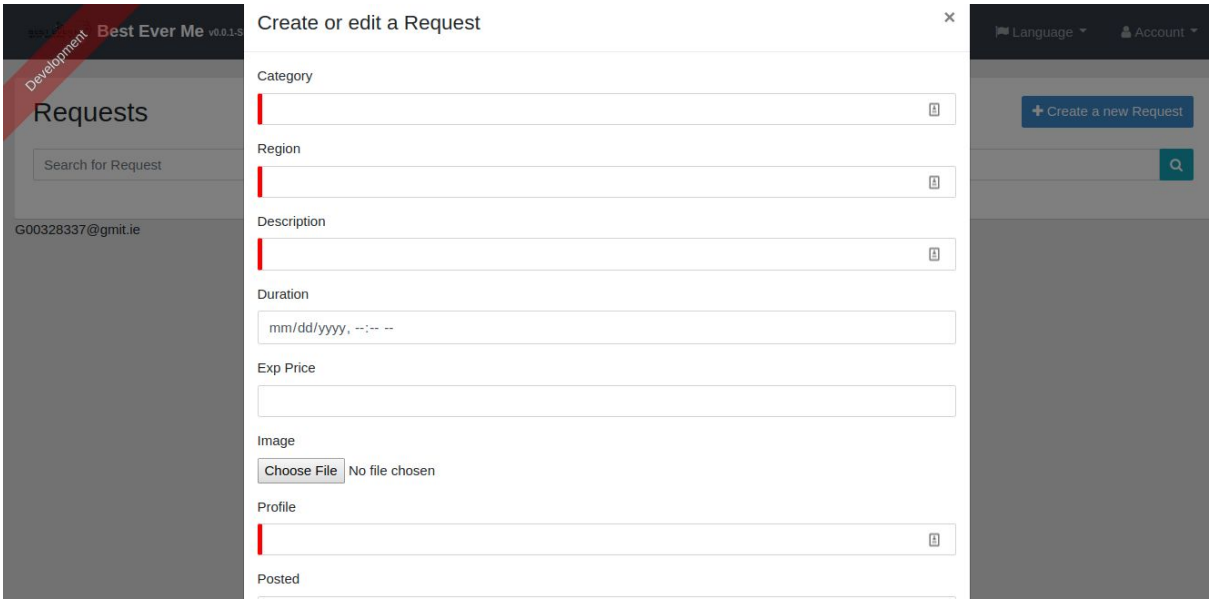


Microservices

This section will deal with microservices already implemented and incorporated into BEM platform. These microservices only contain server side java code and front end functionality is provided by web interface in gateway web application and Ionic based mobile applications. All these service have their own access to the database separately, but for the development purposes all microservice as well as gateway are using the same database running inside the Docker container.

Request Service

This microservice application is responsible for reverse auction services. It accepts clients request for service. It is here where all the information is stored and processed. Including Category by which Service providers can monitor client's requests and respond to them if they wish.

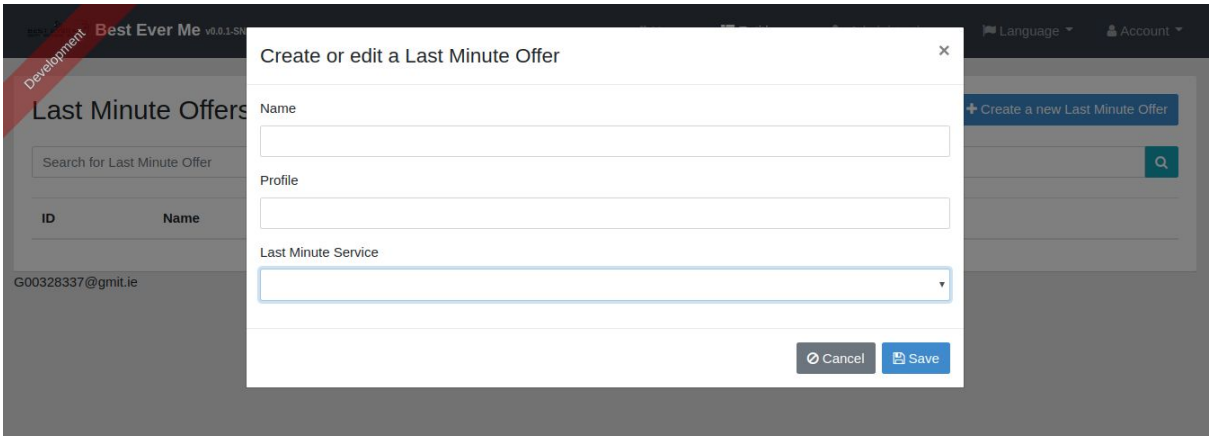


Last Minute Service

This is the interface for the service providers advertise Last-minute offers to an clients. Clients are subscribed to these offers by categories their choose.

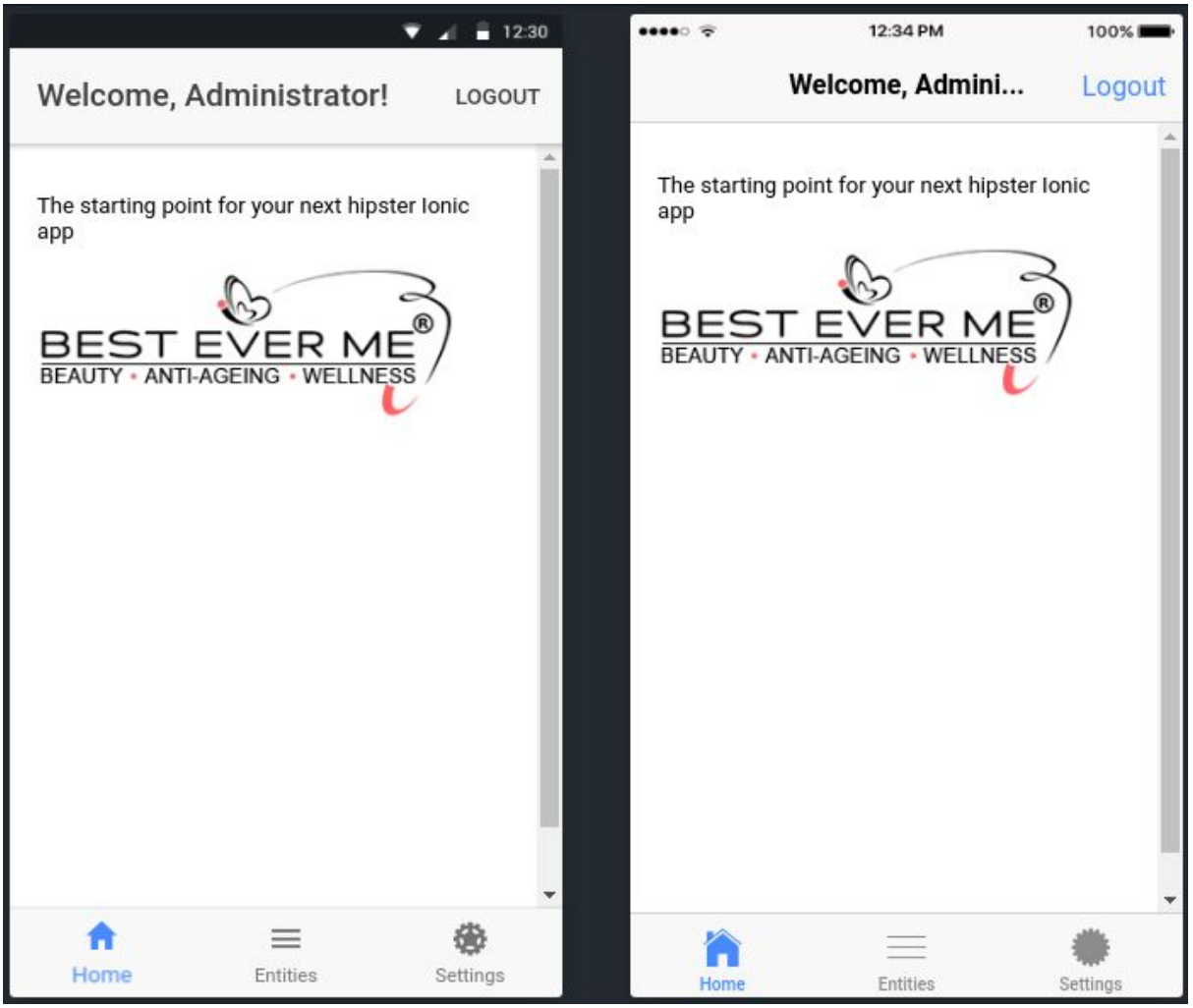
Client side for Last-minute offer

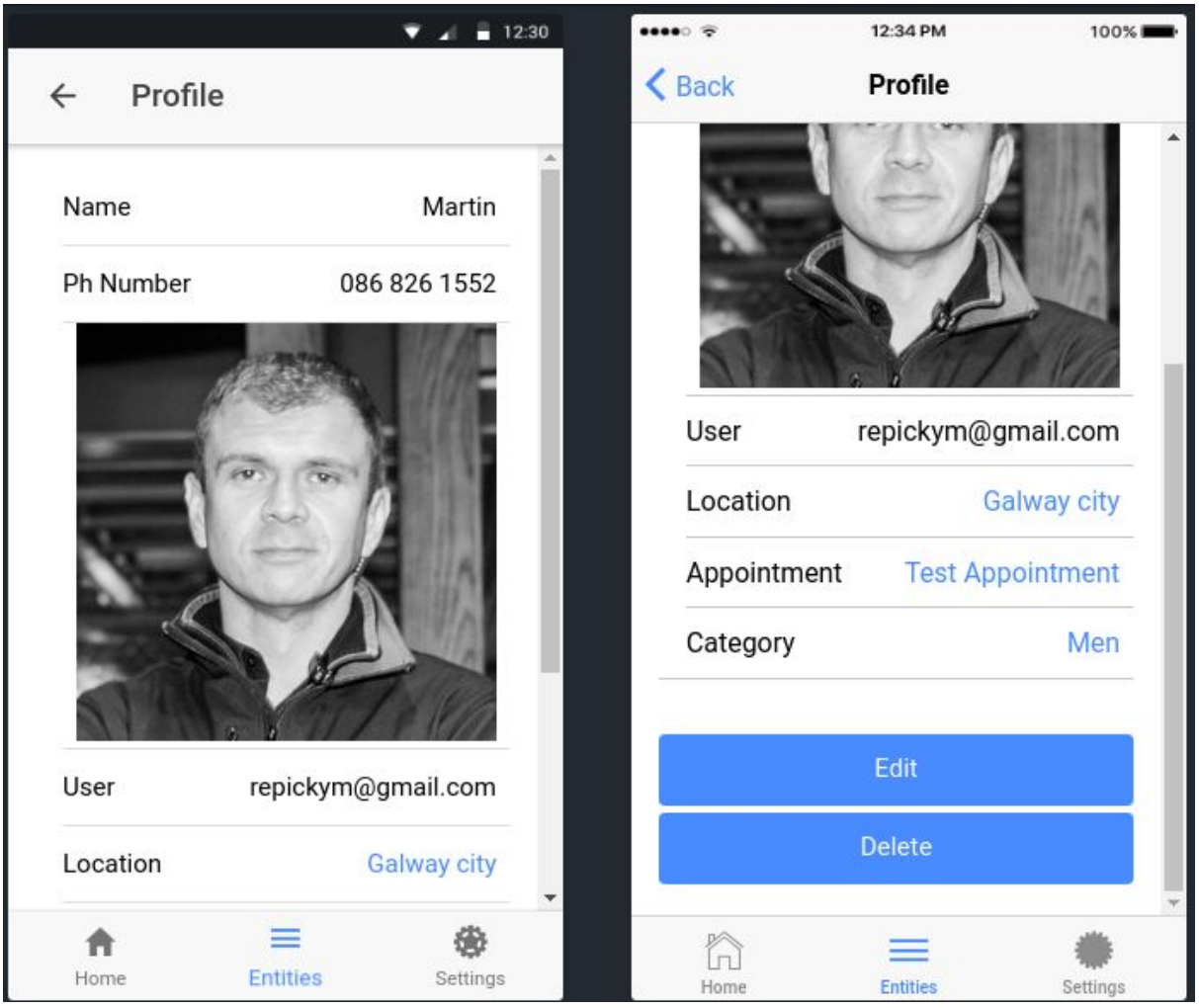
This is an interface for clients to accept service providers Last-minute offers. Once the offer is made the appointment for both Service provider and client is created.



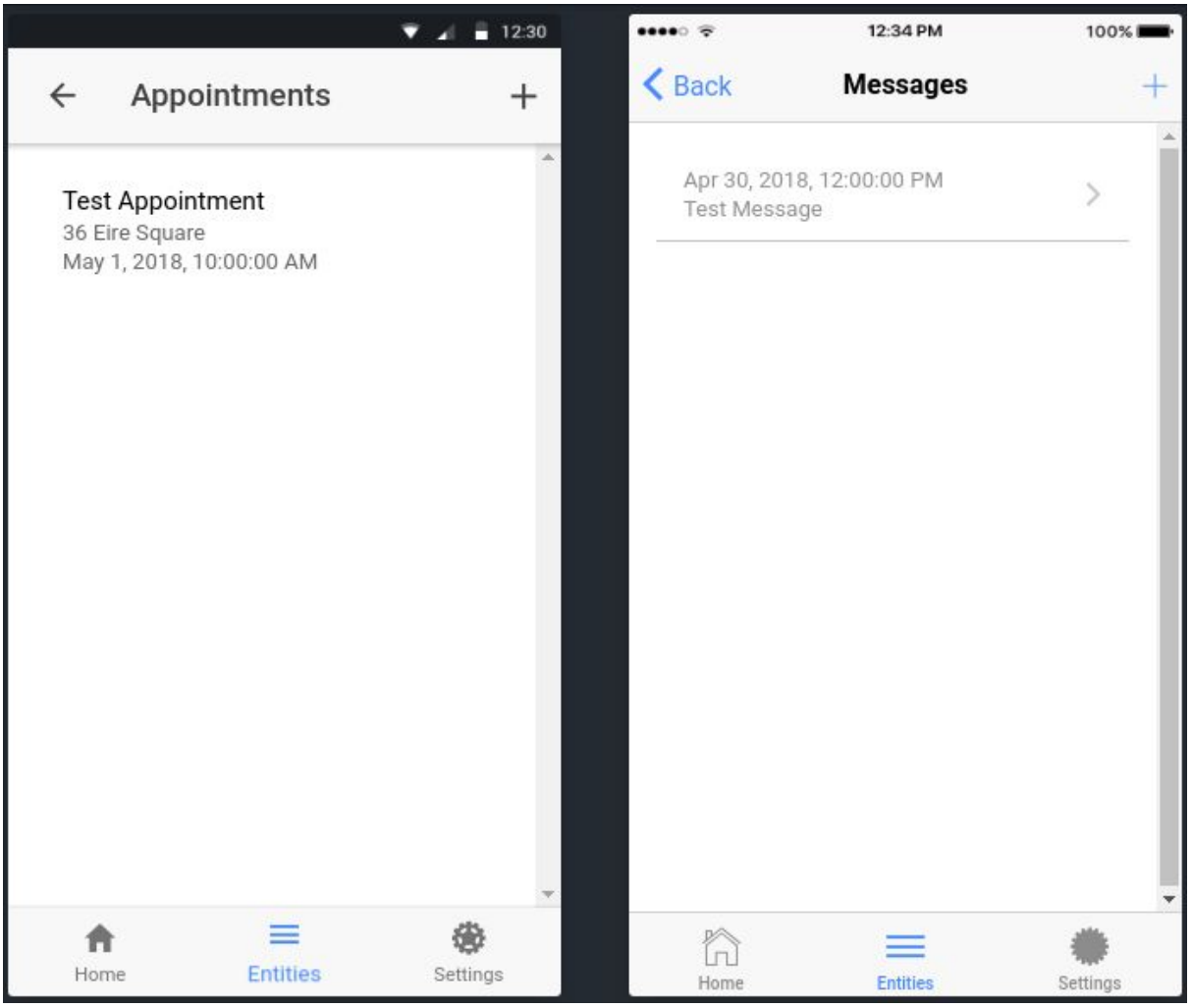
Ionic Reverse Auction Mobile Application

In Cross platform Ionic based mobile application all the non administrative functionality described in the section above are available to both clients and service providers. The user is greeted with home page where their can decide interact with their profile/s.

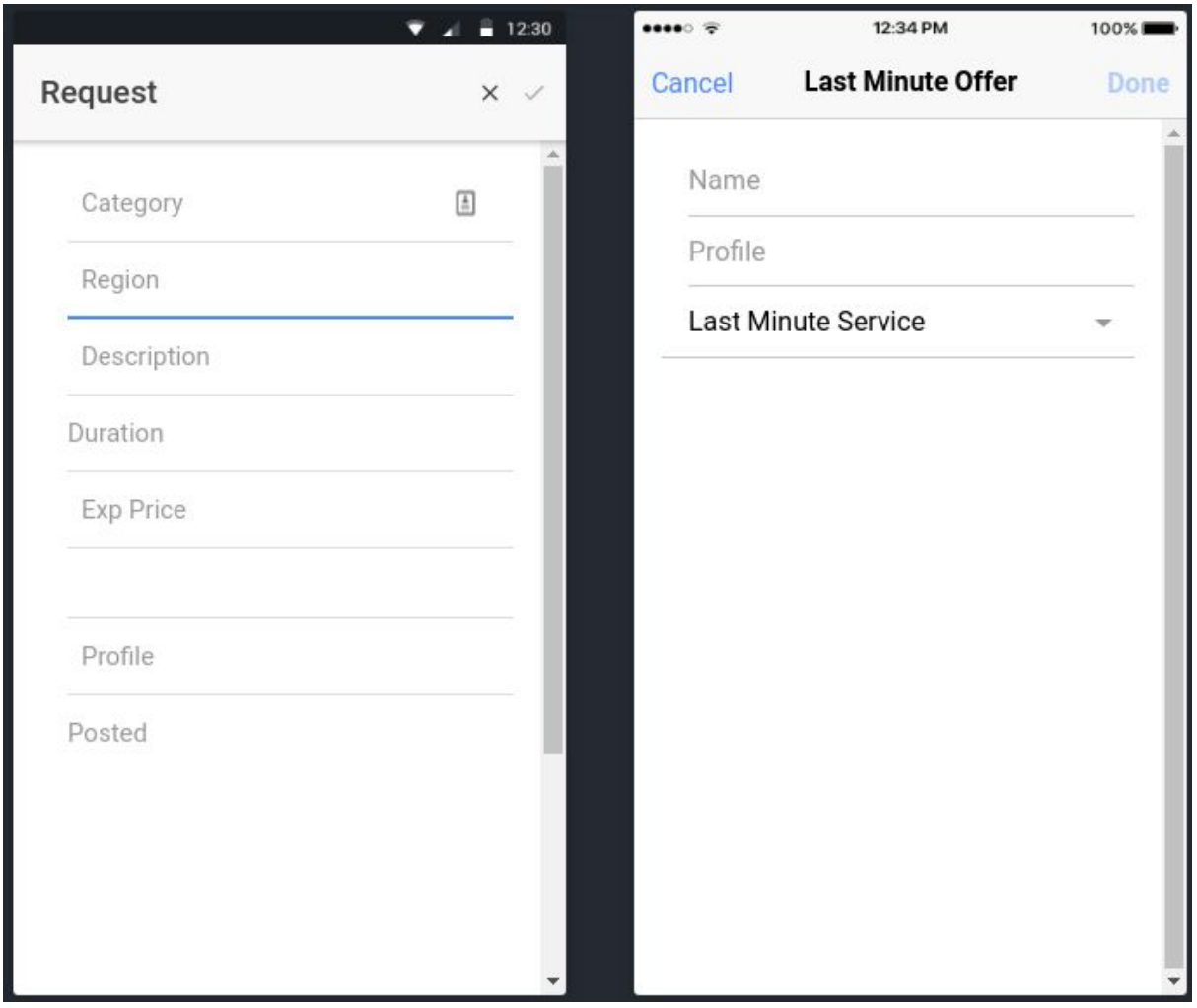


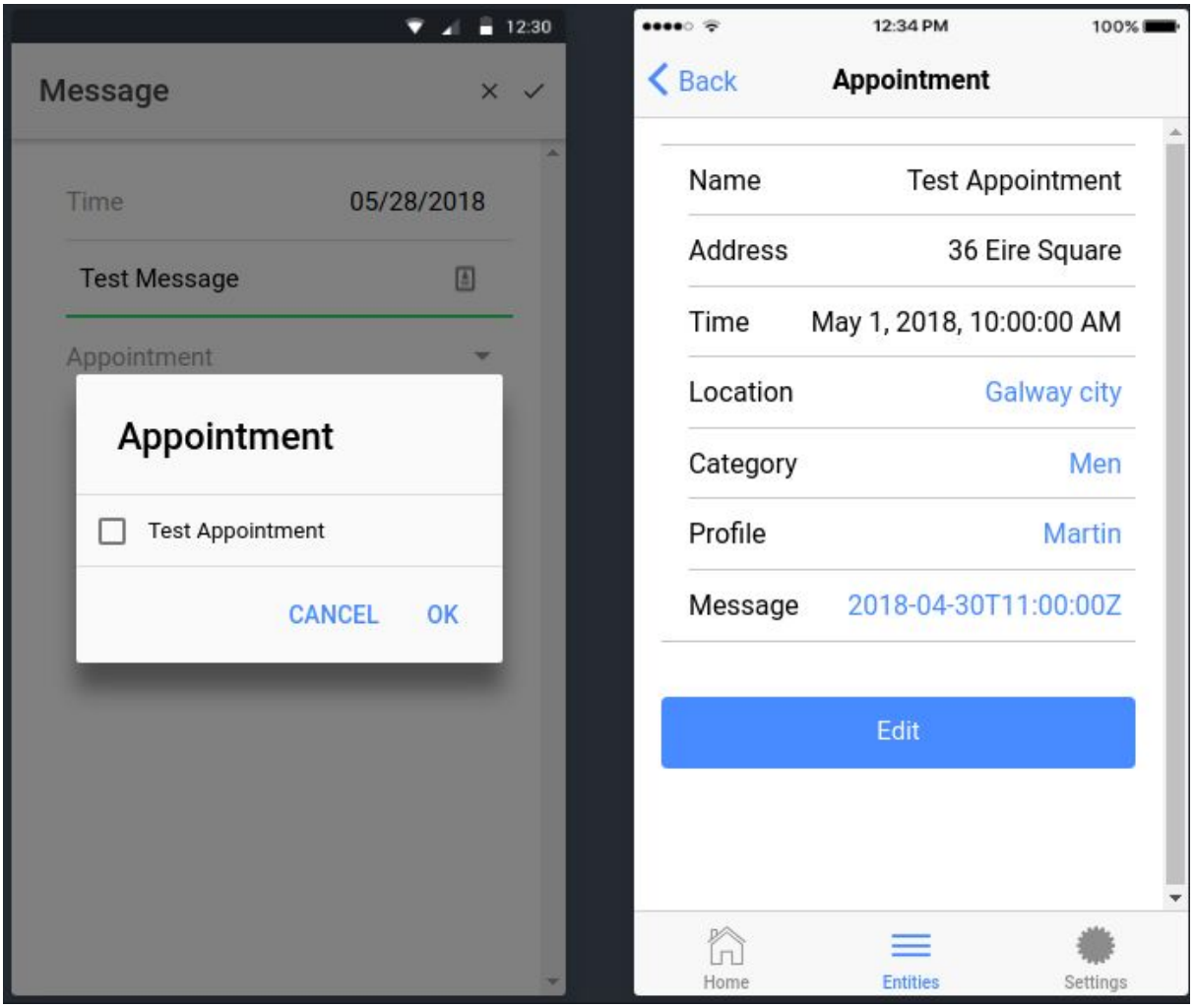

12:30
Profile
Name
Martin
Ph Number
086 826 1552
User
repickym@gmail.com
Location
Galway city
Home
Entities
Settings
12:34 PM
100%
Back
Profile
User
repickym@gmail.com
Location
Galway city
Appointment
Test Appointment
Category
Men
Edit
Delete
Home
Entities
Settings

The list of appointments is available to each profile as well all the messages associated with each appointment.



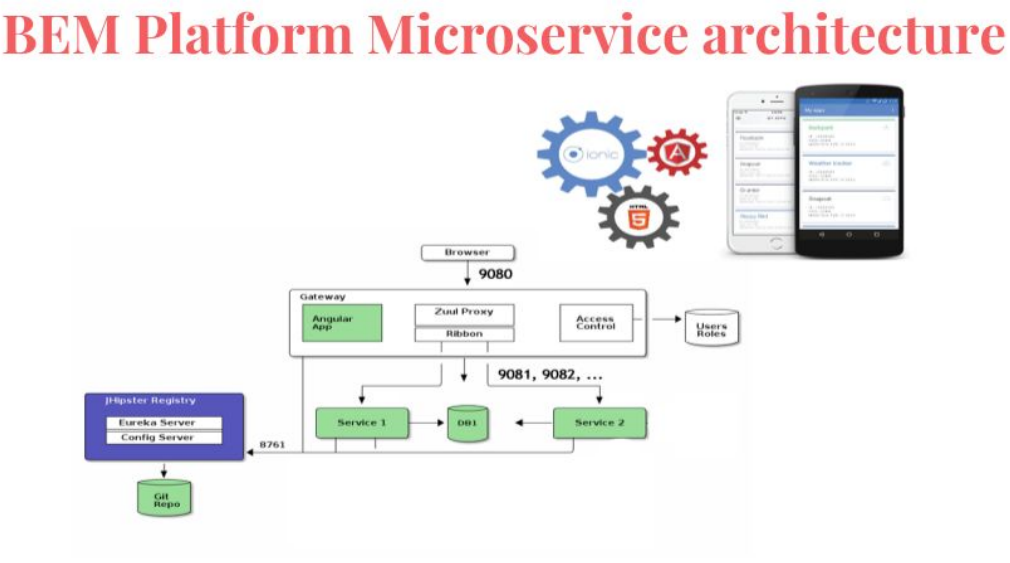
User can create Last minute Offer as well Requests.


12:30
Request
Category
Region
Description
Duration
Exp Price
Profile
Posted
12:34 PM
100%
Cancel
Last Minute Offer
Done
Name
Profile
Last Minute Service


12:30
Message
Time
05/28/2018
Test Message
Appointment
Appointment
Test Appointment
CANCEL
OK
12:34 PM
100%
Back
Appointment
Name
Test Appointment
Address
36 Eire Square
Time
May 1, 2018, 10:00:00 AM
Location
Galway city
Category
Men
Profile
Martin
Message
2018-04-30T11:00:00Z
Edit
Home
Entities
Settings

System Design

Best Ever Me Platform is based on Microservice architecture. Gateway application is providing routing, while JHipster Registry is providing Eureka server for discovery service and Spring Cloud Config Server to provide runtime configuration to all applications. All the application are capable of running their separate database queries and do so over API. In development mode all the services are using single instance of MariaDb.



Database schema listed below describe tables and their links.
Gateway schema:

- User is linked to Profile by One to Many relationship
- Category is linked to Profile by Many to One relationship
- Location is linked to Profile by One to One relationship
- Appointment is linked to Profile by Many to One relationship
- Message is linked to Appointment by Many to One relationship

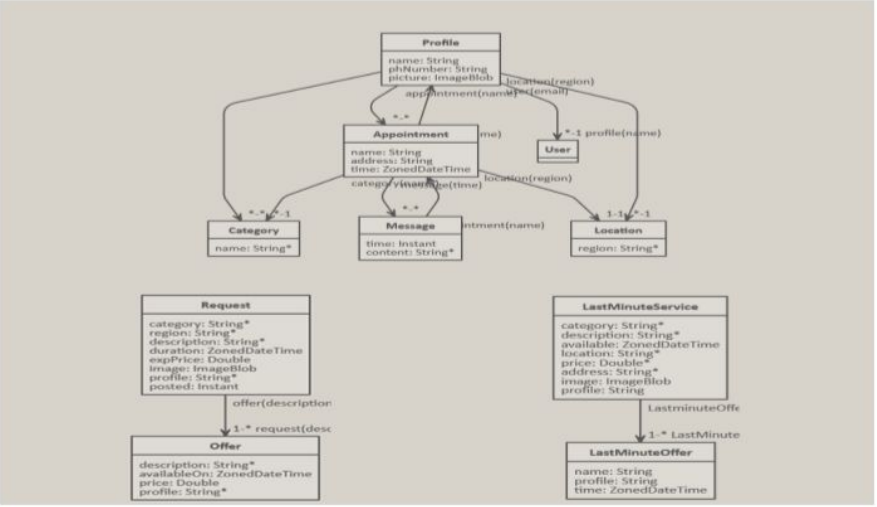
Request schema:

- Offer is linked to Request by Many to One relationship

Last-Minute-Service schema:

- LastMinuteOffer is linked to LastMinuteServie by One to One reationship

Microservices Schema



Technology Review

In this section a brief description of technology used in this project will be provided. The decision was made to use open source technology as much as possible throughout this project.

Database

MariaDB

MariaDB is a community-developed fork of the MySQL relational database management system intended to remain free under the GNU GPL. Development is led by some of the original developers of MySQL, who forked it due to concerns over its acquisition by Oracle Corporation. MariaDB intends to maintain high compatibility with MySQL, ensuring a drop-in replacement capability with library binary equivalency and exact matching with MySQL APIs and commands. This led to make a decision to use MariaDB in all of the projects applications.

Ehcache

Ehcache is an open source Java distributed cache for general purpose caching, Java EE and light-weight containers. Ehcache is available under an Apache open source license The software is still open source but some new major functionalities (Fast Restartability Consistency) are available only in commercial products like Enterprise Ehcache and BigMemory, which are not open source.

Elasticsearch

Elasticsearch provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License. Elasticsearch is developed alongside a data-collection and log-parsing engine called Logstash, and an analytics and visualisation platform called Kibana. The three products are designed for use

as an integrated solution, referred to as the "Elastic Stack" (formerly the "ELK stack"). In layman terms, it is just a search engine.

Backend

Server side software was developed in Java on Spring framework.

Spring Framework

Spring framework is an open source Java platform. The core features of the Spring Framework can be used in developing any Java application, but there are extensions for building web applications on top of the Java EE platform. Spring framework targets to make J2EE development easier to use and promotes good programming practices by enabling a POJO-based programming model. Spring enables developers to develop enterprise-class applications using POJOs. The benefit of using only POJOs is that you do not need an EJB container product such as an application server but you have the option of using only a robust servlet container such as Tomcat or some commercial product. Spring's web framework is a well-designed web MVC framework, which provides a great alternative to web frameworks such as Struts or other over-engineered or less popular web frameworks.

Spring Security

Spring Security is a framework that focuses on providing both authentication and authorization to Java applications. Like all Spring projects, the real power of Spring Security is found in how easily it can be extended to meet custom requirements.

Spring MVC & REST

Spring MVC supports REST. It is easier to build restful web services with spring with it’s annotation based MVC framework. Spring REST architecture is also based on Spring MVC, slightly making the difference on the View part. Traditional Spring MVC relies on the View technology to render the model data, the Spring REST architecture also does the same, except that the model object is set directly into the HTTP response, which the @ResponseBody converts into JSON/XML automatically. The output of a RESTful web service has to be a JSON or an XML, a standard format that could be easily handled across different consumer application platforms.

Spring Data JPA

Spring Data JPA aims to significantly improve the implementation of data access layers by reducing the effort to the amount that’s actually needed. As a developer you write your repository interfaces, including custom finder methods, and Spring will provide the implementation automatically.

JHipster Registry

The JHipster Registry is a runtime application, provided by the JHipster team. Like the JHipster generator, it is an Open Source, Apache 2-licensed application, and its source code is available on GitHub.
The JHipster Registry has three main purposes:

- It is a an Eureka server, that serves as a discovery server for applications. This is how JHipster handles routing, load balancing and scalability for all applications.It is a Spring Cloud Config server, that provide runtime configuration to all applications.
- It is an administration server, with dashboards to monitor and manage applications.

All those features are packaged into one convenient application with a modern Angular-based user interface.

Frontend

Angular 5

JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. The JavaScript components complement Apache Cordova, a framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.
The AngularJS framework works by first reading the HTML page, which has additional custom tag attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

Sass

Sass is the most mature, stable, and powerful professional grade CSS extension language in the world. Sass is completely compatible with all versions of CSS. We take this compatibility seriously, so that you can seamlessly use any available CSS libraries. Sass boasts more features and abilities than any other CSS extension language out there. The Sass Core Team has worked endlessly to not only keep up, but stay ahead.

Ionic

Ionic is a complete open-source SDK for hybrid mobile app development. The SDK is built on top of AngularJS and Apache Cordova. The more recent releases, known as Ionic 3 or simply "Ionic", are built on Angular. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova.

PhoneGap

PhoneGap Build takes the pain out of compiling PhoneGap apps. Get app-store ready apps without the headache of maintaining native SDKs. Our PhoneGap Build service does the work for you by compiling in the cloud.

Cordova

Cordova wraps your HTML/JavaScript app into a native container which can access the device functions of several platforms. These functions are exposed via a unified JavaScript API, allowing you to easily write one set of code to target nearly every phone or tablet on the market today and publish to their app stores.

Developing tools

JHipster

JHipster provides tools to generate a project with a Java stack on the server side (using Spring Boot) and a responsive Web front-end on the client side (with Angular and Bootstrap). It can also create microservice stack with support for Netflix OSS, Docker and Kubernetes. JHipster is a development platform to generate, develop and deploy Spring Boot + Angular/React Web applications and Spring microservices.

Gradle

Gradle is an open-source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used by Apache Maven for declaring the project configuration.[3] Gradle uses a directed acyclic graph ("DAG") to determine the order in which tasks can be run.
Gradle was designed for multi-project builds, which can grow to be quite large. It supports incremental builds by intelligently determining which parts of the build tree are up to date; any task dependent only on those parts does not need to be re-executed.

Yarn

The fundamental idea of YARN is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons. The idea is to have a global ResourceManager (RM) and per-application ApplicationMaster (AM). An application is either a single job or a DAG of jobs.
The ResourceManager and the NodeManager form the data-computation framework. The ResourceManager is the ultimate authority that arbitrates resources among all the applications in the system. The NodeManager is the per-machine framework agent who is responsible for containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager/Scheduler.
The per-application ApplicationMaster is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to execute and monitor the tasks.

Docker

Docker is a computer program that performs operating-system-level virtualization also known as containerization. It is developed by Docker, Inc. Docker is primarily developed for Linux, where it uses the resource isolation features of the Linux kernel such as cgroups and kernel namespaces, and a union-capable file system such as OverlayFS and others to allow independent "containers" to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines (VMs). The Linux kernel's support for namespaces mostly isolates an application's view of the operating environment, including process trees, network, user IDs and mounted file systems, while the kernel's cgroups provide resource limiting for memory and CPU. All components of this projects are modularized into docker containers and can be run in different environments and deployed on multiple Cloud computing providers like Amazon AWS, Google cloud and Microsoft Azure.

Conclusion

The project as a whole has grown from simple Monolithic application to a Microservices platform with different modularized components. This design was chosen for ability incorporate current business projects as well future projects under the same platform. Creating a Monolithic application would only led to more fragmentation of company's services and need for unified platform was more demanding than competition of mobile application itself. As a result of numerous meetings and consultancies with CEO of Best Ever World ltd Portia Quinn, the decisions was made to offer me a position of CTO in the company which I accepted. We are currently applying for funding via New Frontiers, entrepreneur development, an Enterprise Ireland programme to continue development of Best Ever Me platform and reverse auction cross-platform mobile application. We strongly believe there is demand for such a services in the industry.

This project proof to be more challenging in many ways then anticipated, but not less brought me new perspectives and opportunities. It made me consider other development approaches and architecture designs to meet clients current and future needs. It considerably improved my level of communication with client and understanding the business aspect of software development. This knowledge led me to seek new ways to meet business demands more efficiently in shorter period of time and with limited human resources while utilizing up-to date technologies and methodologies.
As a result my understanding of design patterns and technologies currently used in industry improved considerably, here's a list of technologies that I had to learn and understand without any prior knowledge:

- Microservices architectures with application discovery and load balancing.
- Open API and its Swagger implementation.
- JSON Web Tokens.
- Yarn and Gradle as development tools.

And a list of technologies which I had some prior knowledge but my skills of implementing them improved considerably:

- Spring framework especially Spring security and Spring Cloud config server
- Angular framework including Ionic toolkit with Sass styling
- Docker as development tool, especially Docker compose

The decision to use these technologies was based on their classification as Open source, which is an important aspect to consider for future development and business costs. The use of Web technologies was a logical solution for development of software solutions for multiple platforms as Angular is popular and very powerful framework to create single page websites and most of the code can be reused for mobile development with Ionic toolkit. This can deliver cross-platform mobile applications with the use of Cordova and Phonegap. The decision to use Java Spring framework to provide server side software solution was made due to my strong knowledge in Java language and Spring's well developed tools and libraries for microservices architecture.
All the modules of the project are well documented and modularized by Docker files, this decision was made early in the development and provide flexibility in development and deployment.
The project is currently still in active development and with close cooperation of CEO of the company. The authentication and authorisation components of platform like email authentication and JSON Web Tokens authorisation are well developed. Reverse auction and Last-minute offers services are developed to an extend and lack restrictions on users to some functionalities as well an implementation of Stripe payment system. This was mainly due to the massive size of the project and the lack of time and resources in development. These issues are being currently addressed and with application for funding via New Frontiers which will provide much needed resources for development of a fully working prototype should be a matter of weeks then months. This will lead to a next stage of development where testing with actual client can be performed.

Demo video can be found on YouTube.
Link: https://www.youtube.com/watch?v=xVc7xMLfHl4

All the source code to this project can be found on Github.
Link: https://github.com/MartinRep/BEMmicro