

## Projekt architektury systemu

Nr zespołu: <b>5@KSSR</b>	Skład zespołu: <b>1. Łukasz Dzwonkowski - kierownik 2. Karol Rumianowski 3. Mateusz Nowak 4. Paweł Mazurek</b>
Nazwa projektu: <b>Śledzenie pozycji smartfona, smartfonów</b>	

Nazwa dokumentu: <b>Projekt architektury systemu</b>	Nr wersji: <b>1.0</b>
Odpowiedzialny za dokument: <b>Paweł Mazurek</b>	Data pierwszego sporządzenia: <b>14.06.2014 r.</b>
Przeznaczenie: <b>Projekt architektury systemu aplikacji internetowej</b>	Data ostatniej aktualizacji: <b>14.06.2014 r.</b>

### Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data

### Spis treści

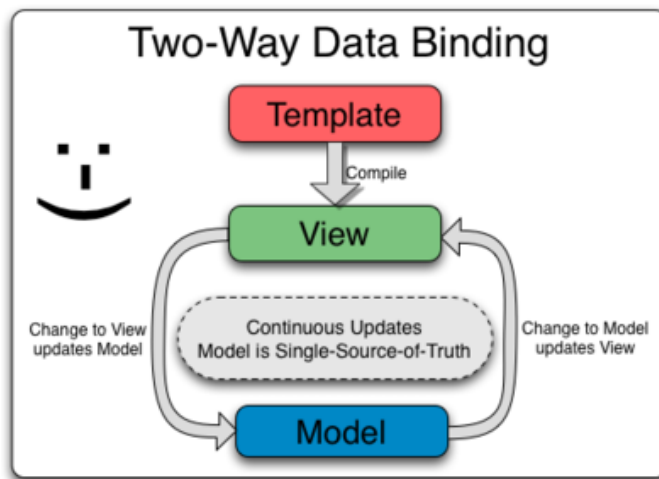
1.	Koncepcja infrastruktury systemu .....	2
2.	Schemat architektury systemu .....	2
3.	Warstwy architektoniczne.....	3
4.	Technologie i biblioteki .....	3

## 1. Koncepcja infrastruktury systemu

*W celu zapewnienia wysokiej jakości kodu, oraz oddzielenia warstw aplikacji, kod aplikacji internetowej podzielony został na 3 warstwy, tj. MVC (model, view, controller).*

*Wykorzystanie frameworku Angular.js wymusza po części taką a nie inną strukturę kodu.*

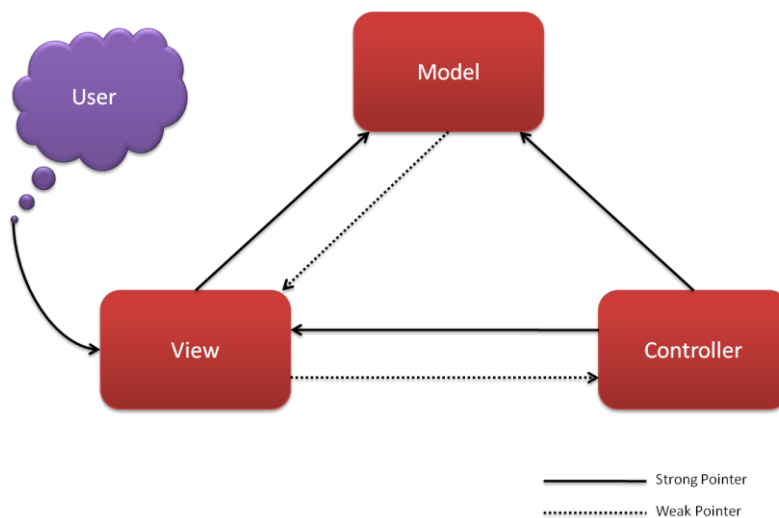
*Za komunikację pomiędzy warstwą model a view odpowiada Angular, udostępniając tzw. Two Way Data-Binding, którego oficjalny schemat wygląda następująco:*



*Czyli zmiana wartości zmiennej w kontrolerze, powoduje jej automatyczną aktualizację w widoku.*

*Za komunikację pomiędzy warstwą controller a model, odpowiada programista, pobierając dane przez protokół HTTP, i zapisując je w pamięci ulotnej przeglądarki, lub np. w 'Session Storage' (rodzaj pamięci trwałej w HTML5, która zastępuje stare 'cookies', lecz dane są usuwane po zamknięciu karty przeglądarki).*

## 2. Schemat architektury systemu



### 3. Warstwy architektoniczne

LAYR_001	<b>Warstwa widoku</b>
Opis:	<i>Widokiem jest kod HTML, oraz opisujący do CSS. Do CSS wykorzystany został LESS, pre-procesor CSS, pozwalający na składnię normalnie niemożliwą do uzyskania w czystym CSS. Np. pozwala na tworzenie zmiennych, lub logicznego grupowania klas</i>

LAYR_002	<b>Warstwa kontroler</b>
Opis:	<i>Kontrolerem są skrypty napisane w Angular.js. Zarządzają one między innymi Data Binding (zmiana wartości zmiennej od razu skutkuje zmianą jej wartości w widoku), logiką biznesową, oraz walidacją formularzy.</i>

LAYR_003	<b>Warstwa model</b>
Opis:	<i>Modelem jest serwer, wysyłane są do niego zapytania, i otrzymywane odpowiedzi, które aplikacja WWW przetwarza. Serwer jest odpowiedzialny za przechowywanie danych w bazie, oraz za operacje na nich. Założenie jest takie, że aplikacja internetowa nie zajmuje się przetwarzaniem danych, udostępnia jedynie interfejs do ich wykorzystania.</i>

### 4. Technologie i biblioteki

LIB_001	Angular.js
Opis:	Framework Javascriptowy SPA (Single page application). Udostępnia wiele narzędzi przydatnych przy tworzeniu dynamicznych, opartych o AJAX stron internetowych

LIB_002	Angular.js – ui.router
Opis:	Pozwala on między innymi na zaimplementowanie możliwości powrotu do poprzedniego widoku kliknięciem na natywny przycisk przeglądarkowy 'Wstecz'.

LIB_003	Ng-boilerplate
Opis:	Szkielet aplikacji pisanej w Angular.js. Udostępnia prekonfigurowane narzędzia pozwalające m.in. na automatyzację typowych zadań programistycznych wykonywanych przez programistę Javascript, bibliotekę Karma służącą do testowania aplikacji, oraz moduł ui.router, wcześniej opisany.

## Projekt architektury systemu

LIB_004	Bootstrap
Opis:	Zapewnia on bazę frontendowych zasobów ponownego użycia, oraz pozwala na implementację systemu grid (responsywny layout).

LIB_004	Less
Opis:	Pre-procesor CSS, pozwalający na składnię normalnie niemożliwą do uzyskania w czystym CSS.

LIB_005	Googe maps
Opis:	Dostawca map