

## WebService w języku Go

### 1. Narzędzia

- a. Go
- b. Framework Beego
- c. REST
- d. Skype
- e. GitHub
- f. Trello
- g. Integracje z zewnętrznym oprogramowaniem:
  - i. AWS S3
  - ii. Draftable Comparison API
  - iii. CloudConvert API
  - iv. Java
  - v. Python

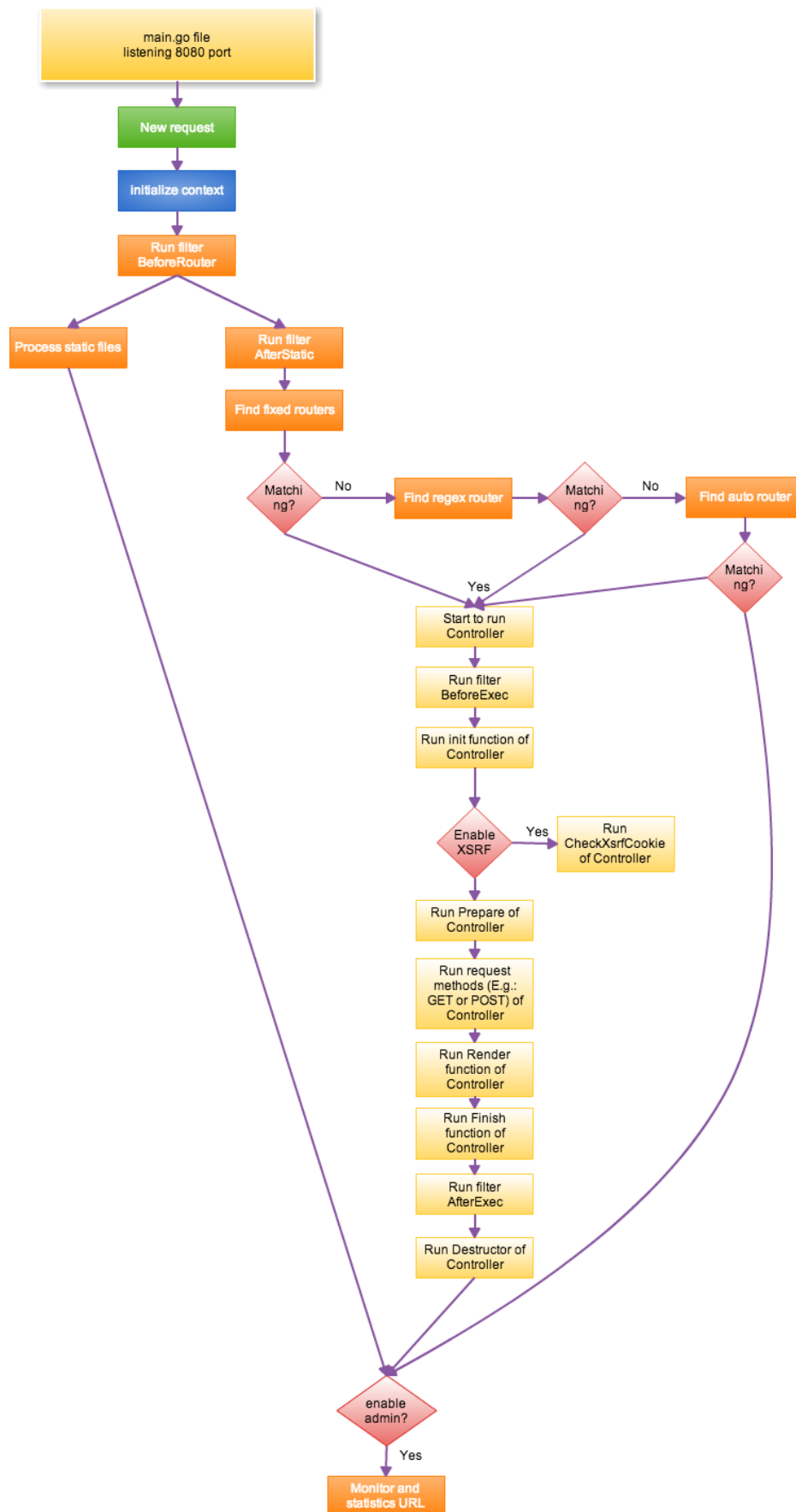
### 2. Analiza dostępności wybranej technologii w danym języku programowania i sposobu jej użytkowania.

Wybrany język programowania: Go

Wybrana technologia: Webservice – REST

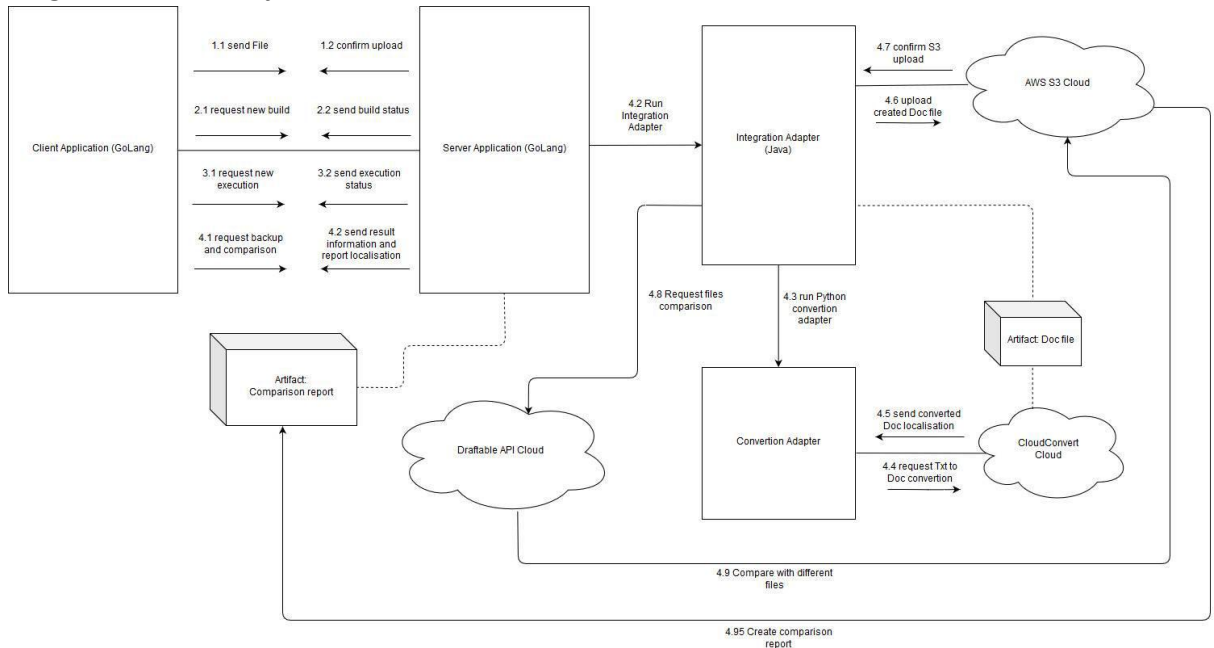
W języku programowania Go występuje mnogość framework'ów, które mogą posłużyć do budowy aplikacji wspierającej API REST'owe. Przykładem mogą służyć następujące: Martini, Gorilla, Gin czy Revel. W celu wybrania najodpowiedniejszego dla naszego projektu, przeszukaliśmy książki [9] czy porównania [11]. Oprócz kwestii czysto technicznych, zależało nam na jasnej, przejrzystej dokumentacji, ze względu na brak doświadczenia i obycia z nimi. Uważamy, że dostępność danej technologii nie sprowadza się jedynie do udostępnienia jej na github'ie, lecz również zadbania o to, by osoby nie mające wcześniej styczności z nią, potrafiły z niej skorzystać. Mając na uwadze powyższe założenia zdecydowaliśmy się na Beego.

Beego wpiera REST oraz model MVC, monitoruje zużycie CPU oraz pamięci. Beego sam w sobie zawiera mnóstwo bibliotek, dzięki czemu nie ma potrzeby szukania oraz dodawania ich do projektu. Wbudowane moduły wspierają kontrolę sesji, cache czy ORM. Beego posiada kilka tutoriali wprowadzających w jego używanie [3].



Rys 1. Diagram ilustrujący obsługę żądania w beego MVC.

### 3. Diagram komunikacji UML



### 4. Wnioski

Projekt okazał się świetną okazją do rozwoju zarówno umiejętności technicznych, jak i miękkich. Co do tych pierwszych, najbardziej oczywistą kwestią jest obycie z językiem Go oraz RESTem. Ponadto zapoznanie się z problemem integracji projektu z chmurą – w tym stosunkowo nowym języku zabrakło bibliotek czy gotowych rozwiązań pozwalających na szybką analizę dostarczanych programów. Dzięki temu nauczyliśmy się, jak radzić sobie z brakiem gotowego rozwiązania, oczywiście innym niż własna implementacja rozwiązania problemu. Z kolei jeśli chodzi o umiejętności miękkie, pewnym wyzwaniem była organizacja całego przedsięwzięcia. Niełatwym zadaniem zebrać jest cztery osoby pracujące w różnych firmach, na różnych stanowiskach i do tego studiujących, by wspólnie wykonać tak duży projekt. Pomocą okazała się tu być kanbanowa tablica z przypisanymi zadaniami, jasne wyznaczenie terminów, rozdzielanie zadań oraz ustalenie kilka tygodni w przód terminów spotkań, na których dzieliliśmy się postępami, wątpliwościami i wiedzą.

W wyniku powstał projekt, który spełnia założenia podane na początku semestru. Diagram komunikacji w czytelny sposób pokazuje schemat działania programu. Zaangażowanie każdego z członków zespołu było okazane w różny, indywidualny sposób. Naturalnie wyłoniły się role, w ramach których każdy mógł zająć się tym, co potrafi najlepiej. Brak popularności technologii kazał nam zmierzyć się z nowym problemem, zapoznać się z nieużywanymi wcześniej narzędziami. Projekt pokazał, że mimo początkowego zagubienia i strachu, dobrze zgrany zespół potrafi przezwyciężyć wszystkie trudności.

#### Źródła:

1. <https://golang.org/doc/> Oficjalna strona języka GO, uzyskano dostęp: 18-01-2019
2. <https://www.pluralsight.com/courses/go> Kurs dotyczący języka Go w serwisie Pluralsight, uzyskano dostęp: 18-01-2019
3. <https://www.pluralsight.com/courses/beego-go-web-app-framework> Kurs dotyczący frameworka Beego w serwisie Pluralsight, uzyskano dostęp: 18-01-2019

4. <https://beego.me/> Oficjalna strona frameworka Beego, zawierająca dokumentację, uzyskano dostęp: 18-01-2019
5. Varghese, S.: Go Recipes: A Problem-Solution Approach, wydanie 1, Apress 2016, ISBN: 978-1-4842-1189. Rozdziały: 1. Beginning Go, 2. Go Fundamentals
6. <https://github.com/shijuvar/go-recipes> Przykłady, kod źródłowy opisany w książce: Go Recipes: A Problem-Solution Approach, uzyskano dostęp: 18-01-2019
7. Newmarch, J.: Network Programming with Go, wydanie 1, Apress 2017, ISBN: 978-1-4842-2692-6. Rozdziały 2. Overview of the Go Language, 14. REST
8. <https://github.com/tumregels/Network-Programming-with-Go> Przykłady, kod źródłowy opisany w książce: Network Programming with Go, uzyskano dostęp: 18-01-2019
9. Varghese, S.: Web Development with Go Building Scalable Web Apps and RESTful Services, wydanie 1, Apress 2015, ISBN: 978-1-4842-1053-6. Rozdziały 4. Getting started with Web Development, 9. Building RESTful Services.
10. <https://github.com/apress/web-dev-w-go> kod źródłowy opisany w książce Web Development with Go Building Scalable Web Apps and RESTful Services, uzyskano dostęp: 18-01-2019
11. <https://github.com/diyan/go-web-framework-comparison> porównanie frameworków w języku Go, uzyskano dostęp: 18-01-2019
12. Gulabani, S.: Practical Amazon EC2, SQS, Kinesis, and S3 A Hands-On Approach to AWS, wydanie 1, Apress 2017, ISBN: 978-1-4842-2841-8. Rozdział5: Hands-on Simple Storage Services (S3)
13. <https://draftable.com/comparison-api> Oficjalna strona Draftwable Comparison API, zawierająca dokumentację oraz przykłady, uzyskano dostęp: 18-01-2019
14. <https://cloudconvert.com/api> Oficjalna strona CloudConvert API, zawierająca dokumentację, uzyskano dostęp: 18-01-2019