# Sprint 2 - 25.03.2015

Zaimplementowana pełna komunikacja między komponentami systemu.

## Etap 1.

Stworzenie uniwersalnego dla wszystkich komponentów sposobu wysyłania i obierania wiadomości. Implementacja go wewnątrz wszystkich komponentów.

Usprawnienie działania komponentów i ich interfejsów użytkownika. Wyspecyfikowanie działania komponentów umożliwiającego wykonanie ich docelowych zadań.

## Etap 2.

Utworzenie metod umożliwiających wysyłanie wiadomości między komponentami systemu.

Utworzenie metod potrafiących obsłużyć przychodzące wiadomości.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MsgName | From | To | What’s inside | How to handle |
| Register message | TM CN BCS | CS | 1. Typ komponentu 2. Typ rozwiązywalych probnlemów (tylko node) | Tylko wewnątrz CS   1. Zapisz na liście nadawcę 2. Nadaj mu ID 3. odeślij Register Response |
| CS | BCS |
| Register Response message | CS | TM CN BCS | 1. Nadane ID 2. Timeout na Status Report | 1. Co timeout ślij Status Report |
| Status message | TM CN BCS | CS | 1. Lista wątków wraz ze szczegółami:    1. czym się zajmuje (ID problemu, podproblemu)    2. jak długo nad tym ślęczy    3. problem type | 1. Odeślij No Operation message |
| No Operation message | CS | TM CN BCS | 1. “Lista” backup serwerów | 1. Zaktualizuj listę BCSów |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Error message | CS | Any | UnknownSender | Send Register Message |
| CS | Any | InvalidOperation | Switch to idle / partially idle state |
| CN  TM | CS | ExceptionOccured | Dwa scenariusze:  1. Ponawiamy od samego początku na podstawie danych.  2. Przesyłamy komunkat dalej do CC |
| CS | CC | Wyświetl komunikat i pozwól użyszkodnikowi zdecydować, czy ponowić czy nie (ma wtedy wpisać dane jeszcze raz). |

Utworzenie metod generujących odpowiednie wiadomości.

Stworzenie statycznych klas będących odpowiedzialnymi za generowanie szkieletów wiadomości.

## Podsumowanie.

Pełen sukces.