

# Licklider - Projekt wstępny TIN

Bartłomiej Partyka, Michał Urbański, Krzysztof Blankiewicz, Tomasz Załuska

27 marca 2020

## 1 Spis treści

• 1. Spis treści . . . . .	1
• 2. Treść zadania . . . . .	1
• 4. Przypadki użycia . . . . .	2
• 5. Środowisko . . . . .	2
• 6. Architektura rozwiązania . . . . .	2
• 7. API modułów . . . . .	2
• 8. Listy komunikatów . . . . .	2
• 9. Sposób testowania . . . . .	2
• 10. Prezentacja . . . . .	2
• 11. Podział pracy . . . . .	2
• 12. Harmonogram pracy . . . . .	2
• 13. Repozytorium github . . . . .	2

## 2 Treść zadania

Urządzenia przechowują zmienne statyczne w pamięciach ulotnej i nieulotnej. Zmienne dynamiczne przechowywane są w pamięci ulotnej. Projektowany system komunikacji działa niezależnie od systemu wykrywania zaniku zasilania i zachowywania stanu urządzenia. Aplikacja korzystająca z usług tego systemu potwierdza konsumpcję danych. Celem systemu jest retransmisja niepotwierdzonych danych po wznowieniu zasilania. System używa stosu TCP/IPv6. Zaprojektować API systemu komunikacyjnego. Ponadto, należy zaprojektować moduł do Wireshark umożliwiający wyświetlanie i analizę zdefiniowanych komunikatów. (Być może pomocnym będzie przejrzanie RFC 5326 "Licklider Transmission Protocol").

### **3 Nazwa własna**

;ciekawa nazwa;

### **4 Przypadki użycia**

### **5 Środowisko**

#### **5.1 Systemy operacyjne**

#### **5.2 Języki i biblioteki**

#### **5.3 Narzędzia testowe**

### **6 Architektura rozwiązania**

;ilustrację i opis struktury logicznej systemu (konceptyjnych bloków funkcjonalnych);

### **7 (ewentualnie)API modułów**

### **8 (ewentualnie)Listy komunikatów**

### **9 Sposób testowania**

### **10 Prezentacja**

;scenariusze testów akceptacyjnych;

### **11 Podział pracy**

### **12 Harmonogram pracy**

;minimum 1, zalecane 2 punkty kontrolne dla odbioru częściowych funkcji/modułów projektu;

### **13 Repozytorium github**

Projekt jest dostępny pod adresem: <https://github.com/six-pd/tin-ics>