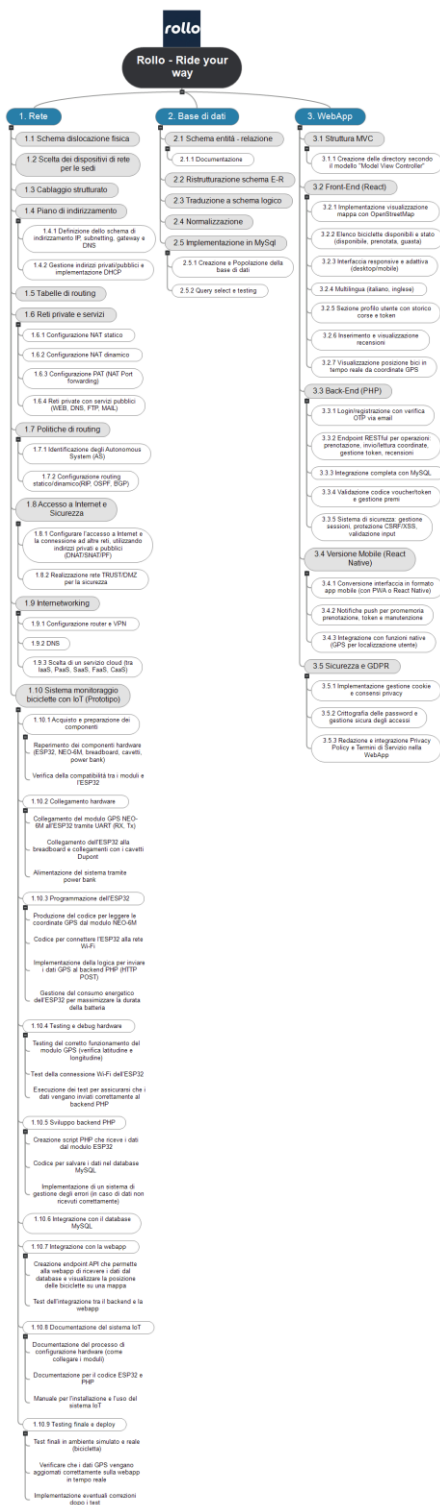


# ROLLO - RIDE YOUR WAY



## 1. Rete 4

1.1 Schema dislocazione fisica .....	4
1.2 Scelta dei dispositivi di rete per le sedi .....	4
1.3 Cablaggio strutturato .....	4
1.4 Piano di indirizzamento .....	4

1.4.1 Definizione dello schema di indirizzamento IP, subnetting, gateway e DNS .....	4
1.4.2 Gestione indirizzi privati/pubblici e implementazione DHCP .....	4
1.5 Tabelle di routing .....	4
1.6 Reti private e servizi .....	4
1.6.1 Configurazione NAT statico .....	4
1.6.2 Configurazione NAT dinamico .....	4
1.6.3 Configurazione PAT (NAT Port forwarding) .....	4
1.6.4 Reti private con servizi pubblici (WEB, DNS, FTP, MAIL).....	5
1.7 Politiche di routing .....	5
1.7.1 Identificazione degli Autonomous System (AS) .....	5
1.7.2 Configurazione routing statico/dinamico(RIP, OSPF, BGP) .....	5
1.8 Accesso a Internet e Sicurezza .....	5
1.8.1 Configurare l'accesso a Internet e la connessione ad altre reti, utilizzando indirizzi privati e pubblici (DNAT/SNAT/PF) .....	5
1.8.2 Realizzazione rete TRUST/DMZ per la sicurezza .....	5
1.9 Internetworking .....	5
1.9.1 Configurazione router e VPN.....	5
1.9.2 DNS.....	5
1.9.3 Scelta di un servizio cloud (tra IaaS, PaaS, SaaS, FaaS, CaaS).....	5
1.10 Sistema monitoraggio biciclette con IoT (Prototipo) .....	5
1.10.1 Acquisto e preparazione dei componenti.....	5
1.10.2 Collegamento hardware .....	6
1.10.3 Programmazione dell'ESP32 .....	6
1.10.4 Testing e debug hardware .....	6
1.10.5 Sviluppo backend PHP .....	7
1.10.6 Integrazione con il database MySQL .....	7
1.10.7 Integrazione con la webapp .....	7
1.10.8 Documentazione del sistema IoT .....	7
1.10.9 Testing finale e deploy.....	8
<b>2. Base di dati</b> .....	<b>8</b>
2.1 Schema entità - relazione .....	8
2.1.1 Documentazione .....	8
2.2 Ristrutturazione schema E-R .....	8
2.3 Traduzione a schema logico .....	8

2.4 Normalizzazione .....	9
2.5 Implementazione in MySql.....	9
2.5.1 Creazione e Popolazione della base di dati .....	9
2.5.2 Query select e testing.....	9
<b>3. WebApp    9</b>	
3.1 Struttura MVC .....	9
3.1.1 Creazione delle directory secondo il modello "Model View Controller" .....	9
3.2 Front-End (React) .....	9
3.2.1 Implementazione visualizzazione mappa con OpenStreetMap .....	9
3.2.2 Elenco biciclette disponibili e stato (disponibile, prenotata, guasta) .....	9
3.2.3 Interfaccia responsive e adattiva (desktop/mobile) .....	9
3.2.4 Multilingua (italiano, inglese) .....	9
3.2.5 Sezione profilo utente con storico corse e token .....	9
3.2.6 Inserimento e visualizzazione recensioni .....	10
3.2.7 Visualizzazione posizione bici in tempo reale da coordinate GPS .....	10
3.3 Back-End (PHP).....	10
3.3.1 Login/registrazione con verifica OTP via email.....	10
3.3.2 Endpoint RESTful per operazioni: prenotazione, invio/lettura coordinate, gestione token, recensioni.....	10
3.3.3 Integrazione completa con MySQL.....	10
3.3.4 Validazione codice voucher/token e gestione premi .....	10
3.3.5 Sistema di sicurezza: gestione sessioni, protezione CSRF/XSS, validazione input .....	10
3.4 Versione Mobile (React Native).....	10
3.4.1 Conversione interfaccia in formato app mobile (con PWA o React Native) .....	10
3.4.2 Notifiche push per promemoria prenotazione, token e manutenzione .....	10
3.4.3 Integrazione con funzioni native (GPS per localizzazione utente) .....	11
3.5 Sicurezza e GDPR.....	11
3.5.1 Implementazione gestione cookie e consensi privacy.....	11
3.5.2 Crittografia delle password e gestione sicura degli accessi .....	11
3.5.3 Redazione e integrazione Privacy Policy e Termini di Servizio nella WebApp .....	11



## **1 1. RETE**

---

### **1.1 1.1 SCHEMA DISLOCAZIONE FISICA**

---

### **1.2 1.2 SCELTA DEI DISPOSITIVI DI RETE PER LE SEDI**

---

### **1.3 1.3 CABLAGGIO STRUTTURATO**

---

### **1.4 1.4 PIANO DI INDIRIZZAMENTO**

---

#### **1.4.1 1.4.1 DEFINIZIONE DELLO SCHEMA DI INDIRIZZAMENTO IP, SUBNETTING, GATEWAY E DNS**

---

#### **1.4.2 1.4.2 GESTIONE INDIRIZZI PRIVATI/PUBBLICI E IMPLEMENTAZIONE DHCP**

---

### **1.5 1.5 TABELLE DI ROUTING**

---

### **1.6 1.6 RETI PRIVATE E SERVIZI**

---

#### **1.6.1 1.6.1 CONFIGURAZIONE NAT STATICO**

---

#### **1.6.2 1.6.2 CONFIGURAZIONE NAT DINAMICO**

---

#### **1.6.3 1.6.3 CONFIGURAZIONE PAT (NAT PORT FORWARDING)**

---

#### **1.6.4 1.6.4 RETI PRIVATE CON SERVIZI PUBBLICI (WEB, DNS, FTP, MAIL)**

---

### **1.7 1.7 POLITICHE DI ROUTING**

---

#### **1.7.1 1.7.1 IDENTIFICAZIONE DEGLI AUTONOMOUS SYSTEM (AS)**

---

#### **1.7.2 1.7.2 CONFIGURAZIONE ROUTING STATICO/DINAMICO(RIP, OSPF, BGP)**

---

### **1.8 1.8 ACCESSO A INTERNET E SICUREZZA**

---

#### **1.8.1 1.8.1 CONFIGURARE L'ACCESSO A INTERNET E LA CONNESSIONE AD ALTRE RETI, UTILIZZANDO INDIRIZZI PRIVATI E PUBBLICI (DNAT/SNAT/PF)**

---

#### **1.8.2 1.8.2 REALIZZAZIONE RETE TRUST/DMZ PER LA SICUREZZA**

---

### **1.9 1.9 INTERNETWORKING**

---

#### **1.9.1 1.9.1 CONFIGURAZIONE ROUTER E VPN**

---

#### **1.9.2 1.9.2 DNS**

---

#### **1.9.3 1.9.3 SCELTA DI UN SERVIZIO CLOUD (TRA IAAS, PAAS, SAAS, FAAS, CAAS)**

---

### **1.10 1.10 SISTEMA MONITORAGGIO BICICLETTE CON IOT (PROTOTIPO)**

---

#### **1.10.1 1.10.1 ACQUISTO E PREPARAZIONE DEI COMPONENTI**

---

#### **1.10.1.1 REPERIMENTO DEI COMPONENTI HARDWARE (ESP32, NEO-6M, BREADBOARD, CAVETTI, POWER BANK)**

---

#### **1.10.1.2 VERIFICA DELLA COMPATIBILITÀ TRA I MODULI E L'ESP32**

---

#### **1.10.2 1.10.2 COLLEGAMENTO HARDWARE**

---

#### **1.10.2.1 COLLEGAMENTO DEL MODULO GPS NEO-6M ALL'ESP32 TRAMITE UART (RX, TX)**

---

#### **1.10.2.2 COLLEGAMENTO DELL'ESP32 ALLA BREADBOARD E COLLEGAMENTI CON I CAVETTI DUPONT**

---

#### **1.10.2.3 ALIMENTAZIONE DEL SISTEMA TRAMITE POWER BANK**

---

#### **1.10.3 1.10.3 PROGRAMMAZIONE DELL'ESP32**

---

#### **1.10.3.1 PRODUZIONE DEL CODICE PER LEGGERE LE COORDINATE GPS DAL MODULO NEO-6M**

---

#### **1.10.3.2 CODICE PER CONNETTERE L'ESP32 ALLA RETE WI-FI**

---

#### **1.10.3.3 IMPLEMENTAZIONE DELLA LOGICA PER INVIARE I DATI GPS AL BACKEND PHP (HTTP POST)**

---

#### **1.10.3.4 GESTIONE DEL CONSUMO ENERGETICO DELL'ESP32 PER MASSIMIZZARE LA DURATA DELLA BATTERIA**

---

#### **1.10.4 1.10.4 TESTING E DEBUG HARDWARE**

---

#### **1.10.4.1 TESTING DEL CORRETTO FUNZIONAMENTO DEL MODULO GPS (VERIFICA LATITUDINE E LONGITUDINE)**

---

#### **1.10.4.2 TEST DELLA CONNESSIONE WI-FI DELL'ESP32**

---

#### **1.10.4.3 ESECUZIONE DEI TEST PER ASSICURARSI CHE I DATI VENGANO INVIATI CORRETTAMENTE AL BACKEND PHP**

---

#### **1.10.5 1.10.5 SVILUPPO BACKEND PHP**

---

##### **1.10.5.1 CREAZIONE SCRIPT PHP CHE RICEVE I DATI DAL MODULO ESP32**

---

##### **1.10.5.2 CODICE PER SALVARE I DATI NEL DATABASE MYSQL**

---

##### **1.10.5.3 IMPLEMENTAZIONE DI UN SISTEMA DI GESTIONE DEGLI ERRORI (IN CASO DI DATI NON RICEVUTI CORRETTAMENTE)**

---

#### **1.10.6 1.10.6 INTEGRAZIONE CON IL DATABASE MYSQL**

---

#### **1.10.7 1.10.7 INTEGRAZIONE CON LA WEBAPP**

---

##### **1.10.7.1 CREAZIONE ENDPOINT API CHE PERMETTE ALLA WEBAPP DI RICEVERE I DATI DAL DATABASE E VISUALIZZARE LA POSIZIONE DELLE BICICLETTE SU UNA MAPPA**

---

##### **1.10.7.2 TEST DELL'INTEGRAZIONE TRA IL BACKEND E LA WEBAPP**

---

#### **1.10.8 1.10.8 DOCUMENTAZIONE DEL SISTEMA IOT**

---

#### **1.10.8.1 DOCUMENTAZIONE DEL PROCESSO DI CONFIGURAZIONE HARDWARE (COME COLLEGARE I MODULI)**

---

#### **1.10.8.2 DOCUMENTAZIONE PER IL CODICE ESP32 E PHP**

---

#### **1.10.8.3 MANUALE PER L'INSTALLAZIONE E L'USO DEL SISTEMA IOT**

---

#### **1.10.9 1.10.9 TESTING FINALE E DEPLOY**

---

#### **1.10.9.1 TEST FINALI IN AMBIENTE SIMULATO E REALE (BICICLETTA)**

---

#### **1.10.9.2 VERIFICARE CHE I DATI GPS VENGANO AGGIORNATI CORRETTAMENTE SULLA WEBAPP IN TEMPO REALE**

---

#### **1.10.9.3 IMPLEMENTAZIONE EVENTUALI CORREZIONI DOPO I TEST**

---

### **2 2. BASE DI DATI**

---

#### **2.1 2.1 SCHEMA ENTITÀ - RELAZIONE**

---

#### **2.1.1 2.1.1 DOCUMENTAZIONE**

---

#### **2.2 2.2 RISTRUTTURAZIONE SCHEMA E-R**

---

#### **2.3 2.3 TRADUZIONE A SCHEMA LOGICO**

---



## 2.4 2.4 NORMALIZZAZIONE

---

## 2.5 2.5 IMPLEMENTAZIONE IN MYSQL

---

### 2.5.1 2.5.1 CREAZIONE E POPOLAZIONE DELLA BASE DI DATI

---

### 2.5.2 2.5.2 QUERY SELECT E TESTING

---

## 3 3. WEBAPP

---

### 3.1 3.1 STRUTTURA MVC

---

#### 3.1.1 3.1.1 CREAZIONE DELLE DIRECTORY SECONDO IL MODELLO "MODEL VIEW CONTROLLER"

---

### 3.2 3.2 FRONT-END (REACT)

---

#### 3.2.1 3.2.1 IMPLEMENTAZIONE VISUALIZZAZIONE MAPPA CON OPENSTREETMAP

---

#### 3.2.2 3.2.2 ELENCO BICICLETTE DISPONIBILI E STATO (DISPONIBILE, PRENOTATA, GUASTA)

---

#### 3.2.3 3.2.3 INTERFACCIA RESPONSIVE E ADATTIVA (DESKTOP/MOBILE)

---

#### 3.2.4 3.2.4 MULTILINGUA (ITALIANO, INGLESE)

---

#### 3.2.5 3.2.5 SEZIONE PROFILO UTENTE CON STORICO CORSE E TOKEN

---

### **3.2.6 3.2.6 INSERIMENTO E VISUALIZZAZIONE RECENSIONI**

---

### **3.2.7 3.2.7 VISUALIZZAZIONE POSIZIONE BICI IN TEMPO REALE DA COORDINATE GPS**

---

## **3.3 3.3 BACK-END (PHP)**

---

### **3.3.1 3.3.1 LOGIN/REGISTRAZIONE CON VERIFICA OTP VIA EMAIL**

---

### **3.3.2 3.3.2 ENDPOINT RESTFUL PER OPERAZIONI: PRENOTAZIONE, INVIO/LETTURA COORDINATE, GESTIONE TOKEN, RECENSIONI**

---

### **3.3.3 3.3.3 INTEGRAZIONE COMPLETA CON MYSQL**

---

### **3.3.4 3.3.4 VALIDAZIONE CODICE VOUCHER/TOKEN E GESTIONE PREMI**

---

### **3.3.5 3.3.5 SISTEMA DI SICUREZZA: GESTIONE SESSIONI, PROTEZIONE CSRF/XSS, VALIDAZIONE INPUT**

---

## **3.4 3.4 VERSIONE MOBILE (REACT NATIVE)**

---

### **3.4.1 3.4.1 CONVERSIONE INTERFACCIA IN FORMATO APP MOBILE (CON PWA O REACT NATIVE)**

---

### **3.4.2 3.4.2 NOTIFICHE PUSH PER PROMEMORIA PRENOTAZIONE, TOKEN E MANUTENZIONE**

---

### **3.4.3 3.4.3 INTEGRAZIONE CON FUNZIONI NATIVE (GPS PER LOCALIZZAZIONE UTENTE)**

---

## **3.5 3.5 SICUREZZA E GDPR**

---

### **3.5.1 3.5.1 IMPLEMENTAZIONE GESTIONE COOKIE E CONSENSI PRIVACY**

---

### **3.5.2 3.5.2 CRITTOGRAFIA DELLE PASSWORD E GESTIONE SICURA DEGLI ACCESSI**

---

### **3.5.3 3.5.3 REDAZIONE E INTEGRAZIONE PRIVACY POLICY E TERMINI DI SERVIZIO NELLA WEBAPP**

---