### 1. Introduction

#### 1.1 Purpose

Global Track is a web-based shipment tracking system designed to provide real-time tracking capabilities for logistics and shipping companies. The system allows users to create, track, and manage shipments while providing notifications to both shippers and receivers.

#### 1.2 Scope

The system includes features for:

- User authentication and authorization

- Shipment creation and management

- Real-time tracking

- Email notifications

- Admin dashboard

- Contact form

### 2. System Overview

#### 2.1 System Architecture

- Frontend: React.js with TypeScript

- Backend: Firebase (Firestore, Authentication)

- Email Service: EmailJS

- Hosting: Firebase Hosting

#### 2.2 User Classes

1. \*\*Public Users\*\*

- Can track shipments

- Can submit contact forms

- Can view public information

2. \*\*Registered Users\*\*

- All public user capabilities

- Can create shipments

- Can receive notifications

3. \*\*Administrators\*\*

- All registered user capabilities

- Can manage all shipments

- Can access admin dashboard

- Can manage user accounts

### 3. Functional Requirements

#### 3.1 User Authentication

- User registration

- User login

- Password reset

- Email verification

#### 3.2 Shipment Management

- Create new shipments

- Generate unique tracking numbers

- Update shipment status

- View shipment history

- Delete shipments (admin only)

#### 3.3 Tracking System

- Real-time status updates

- Location tracking

- Estimated delivery dates

- Package details

#### 3.4 Notification System

- Email notifications for:

- Shipment creation

- Status updates

- Delivery confirmation

- Customizable notification preferences

#### 3.5 Admin Dashboard

- User management

- Shipment overview

- Analytics and reporting

- System settings

### 4. Non-Functional Requirements

#### 4.1 Performance

- Page load time < 3 seconds

- Real-time updates < 1 second

- Support for 1000+ concurrent users

#### 4.2 Security

- Secure authentication

- Data encryption

- Role-based access control

- Input validation

#### 4.3 Reliability

- 99.9% uptime

- Automated backups

- Error logging and monitoring

#### 4.4 Usability

- Responsive design

- Intuitive navigation

- Mobile-friendly interface

- Accessibility compliance

### 5. Technical Requirements

#### 5.1 Frontend

- React.js 18+

- TypeScript 4+

- Material-UI or Tailwind CSS

- React Router

- EmailJS

#### 5.2 Backend

- Firebase 9+

- Firestore

- Firebase Authentication

- Firebase Hosting

#### 5.3 Development Tools

- Git for version control

- ESLint for code quality

- Prettier for code formatting

- Jest for testing

### 6. Free Hosting Options

#### 6.1 Frontend Hosting

1. \*\*Firebase Hosting (Recommended)\*\*

- Free tier includes:

- 10GB storage

- 360MB/day bandwidth

- SSL certificate

- Custom domain support

- Steps to deploy:

```bash

npm install -g firebase-tools

firebase login

firebase init

firebase deploy

```

2. \*\*Netlify\*\*

- Free tier includes:

- 100GB bandwidth/month

- SSL certificate

- Continuous deployment

- Custom domain support

3. \*\*Vercel\*\*

- Free tier includes:

- Unlimited personal projects

- SSL certificate

- Automatic deployments

- Custom domain support

#### 6.2 Backend Services

1. \*\*Firebase (Already using)\*\*

- Free tier includes:

- 1GB Firestore storage

- 50,000 reads/day

- 20,000 writes/day

- Authentication for 50,000 users

2. \*\*EmailJS\*\*

- Free tier includes:

- 200 emails/month

- 2 email templates

- Basic support

### 7. Deployment Process

1. \*\*Frontend Deployment\*\*

```bash

# Build the project

npm run build

# Deploy to Firebase

firebase deploy

```

2. \*\*Environment Setup\*\*

- Create `.env` file with required variables

- Set up Firebase configuration

- Configure EmailJS credentials

3. \*\*Domain Setup\*\*

- Purchase domain (optional)

- Configure DNS settings

- Set up SSL certificate

### 8. Maintenance and Support

#### 8.1 Regular Maintenance

- Weekly code updates

- Monthly security patches

- Quarterly feature updates

#### 8.2 Monitoring

- Firebase Analytics

- Error tracking

- Performance monitoring

#### 8.3 Backup Strategy

- Daily Firestore backups

- Weekly full system backups

- Monthly archive backups

### 9. Future Enhancements

- Mobile application

- API integration capabilities

- Advanced analytics

- Multi-language support

- Payment integration

- Advanced reporting

- Mobile app notifications

- API documentation