TaxiTap

Technical Installation Manual

Revolutionizing South Africa's Minibus Taxi Industry

Version 1.0
August 19, 2025

Contents

1 Introduction **2**
1.1 System Architecture 2

2 Prerequisites **2**
2.1 System Requirements 2
2.2 Required Software 2
2.2.1 Node.js (v18.0.0 or higher) 2
2.2.2 Expo CLI (v6.3.0) 3
2.2.3 Git (Latest Stable) 3
2.2.4 Development Tools 3
2.2.5 Package Manager 3

3 Installation **4**
3.1 Repository Cloning 4
3.2 Dependency Installation 4
3.3 Environment Configuration 4
3.3.1 Convex Backend Setup 4
3.3.2 Environment Variables 5
3.4 Platform-Specific Setup 5
3.4.1 Android Setup 5
3.4.2 iOS Setup (macOS only) 5

4 Deployment and Running **5**
4.1 Development Environment 5
4.1.1 Starting the Development Server 5
4.2 Backend Deployment 6
4.2.1 Convex Backend 6
4.3 Production Deployment 6
4.3.1 Building for Production 6

5 Testing and Verification **7**
5.1 Running Tests 7
5.2 Verification Checklist 7

6 Troubleshooting **7**
6.1 Common Issues and Solutions 7
6.1.1 Node.js Version Conflicts 7
6.1.2 Expo Go Connection Issues 8
6.1.3 Android Build Failures 8
6.1.4 Convex Authentication Issues 8
6.1.5 Platform Directory Issues 8

7 Support and Resources **8**
7.1 Additional Resources 8

1 Introduction

TaxiTap is a revolutionary mobile platform designed to transform South Africa's minibus taxi industry. Our solution bridges the gap between traditional taxi operations and modern technology, creating a seamless experience for both passengers and operators.

1.1 System Architecture

The TaxiTap system comprises the following key components that require installation and configuration:

- **Frontend Mobile Application**: Built with React Native and Expo for cross-platform compatibility (Android and iOS)
- **Backend Services**: Powered by Convex TypeScript serverless architecture for real-time data processing
- **Database Layer**: Convex document-oriented database with real-time synchronization capabilities
- **Cloud Infrastructure**: Hosted on Convex Cloud with automatic scaling and deployment

The installation process involves setting up the development environment, configuring the backend services, and deploying both frontend and backend components. This manual provides detailed instructions for multiple operating systems to ensure broad compatibility.

2 Prerequisites

Before beginning the installation process, ensure your system meets the following requirements and has the necessary software installed.

2.1 System Requirements

Minimum System Requirements

- **Operating System**: Windows 10+, macOS 10.15+, or Ubuntu 18.04+
- **RAM**: 8GB minimum, 16GB recommended
- **Storage**: 10GB free space minimum
- **Internet**: Stable broadband connection required

2.2 Required Software

2.2.1 Node.js (v18.0.0 or higher)

Installation Instructions:

1. Visit https://nodejs.org/
2. Download the LTS version (v18.19.0 or higher)
3. Run the installer and follow the setup wizard
4. Verify installation:

```
node --version
npm --version
```

Expected output: Node.js v18.19.0+ and npm 9.0.0+

2.2.2 Expo CLI (v6.3.0)

Install globally via npm:

```
npm install -g @expo/cli@6.3.0
```

Verify installation:

```
expo --version
```

2.2.3 Git (Latest Stable)

Windows:

1. Download from https://git-scm.com/download/win
2. Run installer with default settings

macOS:

```
brew install git
```

Ubuntu/Linux:

```
sudo apt update
sudo apt install git
```

2.2.4 Development Tools

For Android Development:

- Android Studio (2023.1.1 or higher)
- Android SDK Platform-Tools
- Android SDK Build-Tools (34.0.0)
- Java Development Kit (JDK 11)

For iOS Development (macOS only):

- Xcode (15.0 or higher)
- Xcode Command Line Tools
- iOS Simulator

2.2.5 Package Manager

Either npm (included with Node.js) or Yarn (v1.22.0+):

```
# Optional: Install Yarn
npm install -g yarn@1.22.19
```

3 Installation

3.1 Repository Cloning

1. Open your terminal/command prompt
2. Navigate to your desired development directory
3. Clone the TaxiTap repository:

```
git clone https://github.com/COS301-SE-2025/TaxiTap.git
cd TaxiTap
```

3.2 Dependency Installation

Install all required project dependencies:

```
# Using npm
npm install

# OR using Yarn
yarn install
```

This will install all dependencies specified in `package.json`, including:

- React Native and Expo SDK
- TypeScript and related type definitions
- Convex client libraries
- Navigation and UI components
- Development and testing utilities

3.3 Environment Configuration

3.3.1 Convex Backend Setup

1. Install Convex CLI globally:

   ```
   npm install -g convex@1.16.4
   ```

2. Initialize Convex in your project:

   ```
   npx convex dev --configure
   ```

3. Follow the prompts to:
   - Create a Convex account (if needed)
   - Set up a new project
   - Configure authentication

3.3.2 Environment Variables

Create a `.env` file in the project root:

```
# Convex Configuration
CONVEX_DEPLOYMENT=your_deployment_url
CONVEX_SITE_URL=https://your_project.convex.site

# Expo Configuration
EXPO_PUBLIC_CONVEX_URL=your_convex_url
EXPO_PUBLIC_APP_ENV=development

# Optional: Additional API Keys
GOOGLE_MAPS_API_KEY=your_google_maps_key
PUSH_NOTIFICATION_KEY=your_push_key
```

3.4 Platform-Specific Setup

3.4.1 Android Setup

1. Ensure Android Studio is installed
2. Set up Android SDK environment variables:

   Windows:

   ```
   set ANDROID_HOME=C:\Users\%USERNAME%\AppData\Local\Android\Sdk
   set PATH=%PATH%;%ANDROID_HOME%\tools;%ANDROID_HOME%\platform-tools
   ```

   macOS/Linux:

   ```
   export ANDROID_HOME=$HOME/Android/Sdk
   export PATH=$PATH:$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools
   ```

3. Create or start an Android Virtual Device (AVD)
4. Enable Developer Options and USB Debugging on physical devices

3.4.2 iOS Setup (macOS only)

1. Install Xcode from the App Store
2. Install Xcode Command Line Tools:

   ```
   xcode-select --install
   ```

3. Accept Xcode license:

   ```
   sudo xcodebuild -license accept
   ```

4. Install iOS Simulator (included with Xcode)

4 Deployment and Running

4.1 Development Environment

4.1.1 Starting the Development Server

1. Navigate to the platform directory:

   ```
   cd platform
   ```

2. Start the Expo development server:

```
npx expo start
```

3. For tunnel mode (useful for network connectivity issues):

```
npx expo start --tunnel
```

4. This will display a QR code in the terminal

5. To run on different platforms:
   - **iOS/Android Device:** Scan the QR code with the Expo Go app
   - **Android Emulator:** Press 'a' in the terminal
   - **iOS Simulator:** Press 'i' in the terminal (macOS only)

Running Options

From the platform directory:

- `npx expo start` - Standard development mode
- `npx expo start --tunnel` - Tunnel mode for network issues
- Press 'a' - Launch Android emulator
- Press 'i' - Launch iOS simulator (macOS only)
- Scan QR code - Run on physical device with Expo Go

4.2 Backend Deployment

4.2.1 Convex Backend

1. Deploy functions to Convex:

```
npx convex deploy
```

2. Push database schema:

```
npx convex deploy --schema-only
```

4.3 Production Deployment

4.3.1 Building for Production

Android APK/AAB:

```
expo build:android -t apk
expo build:android -t app-bundle
```

iOS IPA:

```
expo build:ios
```

Web Build:

```
expo export:web
```

5 Testing and Verification

5.1 Running Tests

Execute the test suite to verify installation and functionality:

```
npm run test:frontend
npm run test:backend
npm run test:integration
```

Test Suite Overview

- **Frontend Tests:** Component testing, UI interactions, navigation
- **Backend Tests:** API endpoints, database operations, business logic
- **Integration Tests:** End-to-end workflows, system integration

5.2 Verification Checklist

Installation Verification

Node.js and npm versions correct

Expo CLI installed and accessible

Repository cloned successfully

Dependencies installed without errors

Convex backend configured and deployed

Environment variables set correctly

`npx expo start` runs without issues

QR code displays for device scanning

App runs on target platform(s)

Frontend tests pass

Backend tests pass

Integration tests pass

6 Troubleshooting

6.1 Common Issues and Solutions

6.1.1 Node.js Version Conflicts

Use Node Version Manager (nvm) to manage multiple Node.js versions:

```
# macOS/Linux
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash

# Windows
# Download from: https://github.com/coreybutler/nvm-windows
```

```
nvm install 18.19.0
nvm use 18.19.0
```

6.1.2 Expo Go Connection Issues

- Ensure devices are on the same network
- Try tunnel mode: `npx expo start --tunnel`
- Clear cache: `npx expo start -c`

6.1.3 Android Build Failures

- Clean Gradle cache: `cd android && ./gradlew clean`
- Update Android SDK and build tools
- Check ANDROID_HOME environment variable

6.1.4 Convex Authentication Issues

```
npx convex login
npx convex dev --configure
```

6.1.5 Platform Directory Issues

- Ensure you're in the correct project root
- Verify the platform directory exists
- Check all dependencies are installed

7 Support and Resources

7.1 Additional Resources

- **Project Repository:** https://github.com/COS301-SE-2025/TaxiTap
- **Issue Tracker:** GitHub Issues
- **Documentation Wiki:** Project Wiki
- **Expo Documentation:** https://docs.expo.dev/
- **Convex Documentation:** https://docs.convex.dev/
- **React Native Documentation:** https://reactnative.dev/docs/getting-started

TaxiTap Technical Installation Manual v1.0
Revolutionizing South Africa's Minibus Taxi Industry
August 19, 2025