

Muritin Bus Service - Video Presentation Script

Total Duration: 9-10 Minutes

SLIDE 1: Title Slide (30 seconds)

Visual: App logo, title, key features

Script: "Hello everyone! Today, I'm excited to present **Muritin** - মুড়ির টিন - a comprehensive smart bus service application designed to revolutionize public transportation in Bangladesh.

Muritin provides real-time bus tracking, a digital booking system, ratings and feedback, and in-app communication between riders and conductors. Our goal is to make bus travel more efficient, transparent, and user-friendly for everyone involved.

Let's dive into how we're solving real transportation challenges."

SLIDE 2: Problem Statement (30 seconds)

Visual: List of current problems

Script: "Let me start by highlighting the problems we're addressing.

Currently, passengers in Bangladesh face several challenges: There's no way to track buses in real-time, arrival times are completely uncertain, and there's no digital payment or booking system whatsoever.

Communication between riders and conductors is difficult, route planning is confusing, and there's absolutely no feedback mechanism to improve service quality.

These problems create frustration, wasted time, and an overall poor commuting experience. That's exactly what Muritin solves."

SLIDE 3: Our Solution (30 seconds)

Visual: Solution overview

Script: "Muritin is a comprehensive mobile application that brings bus services into the digital age.

We've designed three distinct user roles - Riders, Conductors, and Bus Owners - each with specialized features. The app includes real-time GPS tracking, a digital request and booking system, automated fare calculation, in-app chat with OTP verification for security, and a complete analytics dashboard for owners.

Everything a modern bus service needs, all in one platform."

SLIDE 4: System Architecture (30 seconds)

Visual: Tech stack components

Script: "Let's look at the technical foundation."

We built Muritin using Jetpack Compose with Kotlin for a modern, native Android experience. The backend runs entirely on Firebase Realtime Database for instant data synchronization, with Firebase Authentication ensuring secure user access.

For location services, we integrate Google Maps API, the Places API, and Directions API. We use GeoFire for efficient location-based queries, and MPAndroidChart for beautiful data visualization in the analytics dashboard.

This tech stack gives us real-time capabilities, scalability, and excellent performance."

SLIDE 5: User Roles & Features (45 seconds)

Visual: Three columns showing each role

Script: "Now, let me explain our three user roles and their features."

Riders - or যাত্রী - can search for nearby pickup and drop points within a 2.5 kilometer radius, submit trip requests, track their bus in real-time, chat with the conductor, view their complete trip history, and rate both the trip and conductor.

Conductors - or কন্ডাক্টর - can view all pending requests, accept and manage bookings, verify riders using OTP codes, share their location during trips, confirm fare collection, and view their performance analytics.

Owners - or মালিক - can register and manage multiple buses, add conductors to their fleet, create detailed trip schedules, view comprehensive analytics and revenue reports, monitor conductor ratings, and track overall business performance.

Each role is carefully designed for their specific needs."

SLIDE 6: Trip Request Flow (45 seconds)

Visual: Step-by-step flow diagram

Script: "Let me walk you through how a typical trip works in Muritin."

First, the rider selects their pickup location on the map. The system then automatically shows nearby bus stops within a 2.5 kilometer radius using GeoHash technology. The rider selects their destination stop from available options, and the fare is automatically calculated based on distance or predefined routes.

The request is then sent to all active conductors on that route. When a conductor accepts, a unique 4-digit OTP is generated for security. Real-time tracking immediately begins, showing the rider exactly where the bus is.

When the bus arrives, the conductor verifies the OTP code, the rider boards, and the trip begins. Upon arrival at the destination, the conductor confirms fare collection, and finally, the rider can rate their experience.

This entire flow is seamless and takes just a few taps."

SLIDE 7: Route Management (30 seconds)

Visual: Route configuration features

Script: "Route management is incredibly flexible in Muritin.

Each bus has a defined route with an origin, destination, and multiple stops in between. We use GeoHash with 5-precision for accurate location matching. The system supports bi-directional routes - meaning buses can go from Point A to Point B, and return from B to A, with different stops highlighted for each direction.

Owners can set custom fare matrices for different route segments, but we also have dynamic fare calculation based on actual distance traveled. All routes are visualized on Google Maps, making everything transparent and easy to understand."

SLIDE 8: Real-time Tracking System (30 seconds)

Visual: Map with tracking visualization

Script: "Real-time tracking is one of our most powerful features.

During active trips, the conductor's location is automatically updated every 10 seconds and stored in Firebase Realtime Database. Riders can see the bus location on their map in real-time, along with their pickup point and destination.

Importantly, tracking is only active during scheduled trips to conserve battery and data. The system automatically starts tracking when a schedule begins and stops when it ends. This gives riders peace of mind while being efficient with resources."

SLIDE 9: Schedule Management (30 seconds)

Visual: Schedule creation interface

Script: "Schedule management gives owners complete control over their operations.

Owners create schedules for each bus, specifying the date, start time, end time, and direction - whether the bus is going forward or returning. Conductors can only accept trip requests during their active scheduled hours, preventing unauthorized trips.

The system automatically cleans up expired schedules after 5 days to keep the database efficient. Conductors can view and manage their own schedules, and there's built-in validation to prevent scheduling conflicts. Everything stays organized automatically."

SLIDE 10: Security Features (30 seconds)

Visual: Security elements

Script: "Security is paramount in Muritin.

We use Firebase Authentication with mandatory email verification for all users. Every trip has OTP-based boarding verification with a unique 4-digit code to ensure only legitimate passengers board.

The app has role-based access control - conductors can only see what they need, owners can only manage their own fleet. A conductor must be explicitly assigned to a bus by the owner. When owners register new conductors, they must re-authenticate with their password for extra security.

All data is securely stored in Firebase with proper access rules. We take security seriously at every level."

SLIDE 11: Communication System (30 seconds)

Visual: Chat interface

Script: "Communication is built right into the app.

Riders and conductors can chat in real-time during trips for any coordination needs. The chat is only enabled during active trips - from the moment a request is accepted until 5 days after the trip ends. This gives time for any follow-up but doesn't clutter the system forever.

All messages are synchronized in real-time through Firebase, and each trip has its own conversation thread. We also integrate phone calling for emergencies. Everything needed for smooth communication is right there in the app."

SLIDE 12: Rating & Feedback System (30 seconds)

Visual: Rating interface

Script: "Accountability and quality improvement come through our rating system.

After each completed trip, riders can rate both the conductor and the bus separately on a scale of 0 to 5 stars.

They can also leave detailed text feedback explaining their rating.

The system automatically calculates aggregate ratings for both conductors and buses. Owners can view complete rating histories including the last 10 reviews with reviewer names. This creates accountability and helps identify top performers or areas needing improvement.

Transparency through ratings benefits everyone."

SLIDE 13: Analytics Dashboard (30 seconds)

Visual: Charts and graphs

Script: "Data-driven decisions are possible with our analytics dashboard.

The system tracks daily trip counts and revenue for each bus, displaying everything in beautiful bar charts. Owners can see bus-wise performance metrics at a glance.

Conductors get their own analytics showing their performance over the last 3 days. The owner dashboard shows overall statistics across their entire fleet, with historical data for trend analysis.

All of this happens automatically - owners just log in and see their business performance in real-time. No manual calculations needed."

SLIDE 14: Advanced Features (30 seconds)

Visual: Advanced features list

Script: "We've also included several advanced features that set Muritin apart.

Riders can request early exits or late exits during their trip if plans change, and the fare is automatically adjusted. The system handles multi-seat bookings effortlessly.

There's automatic cleanup of expired data to keep performance optimal. Every user has access to their complete trip history. Account management is comprehensive - users can update their profile, change passwords, and more.

These features make Muritin not just functional, but truly user-friendly for real-world scenarios."

SLIDE 15: Technical Highlights (30 seconds)

Visual: Performance metrics

Script: "Let me highlight some key technical achievements.

For performance, we have auto-refresh mechanisms running every 15 to 30 seconds to keep data current

without draining resources. Database queries are optimized, and location updates are efficient.

For data management, Firebase Realtime Database gives us instant synchronization, GeoFire handles complex location queries efficiently, and automatic cleanup keeps the database lean.

The UI uses Material Design 3 with full Bengali language support for local users. Navigation is intuitive, and the modern Compose UI is beautiful and responsive.

We've built something that's not just functional, but also fast and pleasant to use."

SLIDE 16: Database Schema (30 seconds)

Visual: Database structure

Script: "Our database is carefully structured for efficiency.

The Users table stores role, name, phone number, age, NID for verification, and ownerId for conductors. Buses contain route information, stops, custom fares, fitness certificates, and tax tokens.

Schedules link buses with conductors, storing start and end times with direction. Requests track every detail of a trip - pickup, destination, seats, fare, status, and the OTP code.

Messages are threaded by request, storing sender ID, text, and timestamps. Ratings capture both conductor and bus scores with comments. And ConductorLocations stores latitude, longitude, and timestamp for tracking.

This schema supports all our features while remaining efficient and scalable."

SLIDE 17: Fare Calculation Logic (30 seconds)

Visual: Fare calculation explanation

Script: "Fare calculation is both flexible and transparent.

First, the system checks if there's a predefined fare in the bus's fare matrix for that specific route segment. If not, it falls back to distance-based calculation at 10 Taka per kilometer, with a minimum fare of 20 Taka.

The base fare is then multiplied by the number of seats booked. If the rider requests an early or late exit, the fare is dynamically adjusted based on the new destination.

We use Google Directions API to calculate actual distance traveled. The fare is always updated when the route changes. Everything is calculated automatically, ensuring fair pricing for riders and proper revenue for owners."

SLIDE 18: Request Lifecycle (45 seconds)

Visual: Complete lifecycle flowchart

Script: "Let me show you the complete lifecycle of a request in the system."

It starts as PENDING when the rider creates the request. The system then finds nearby conductors who are within a 30-minute estimated travel time from the pickup point.

When a conductor accepts, the status changes to ACCEPTED and an OTP is generated. After the conductor verifies the OTP, it becomes OTP_VERIFIED, confirming the rider has boarded.

During the trip, it's marked as IN_TRANSIT. When they reach the destination, it changes to ARRIVED. After the conductor confirms fare collection, it's FARE_COLLECTED. Finally, when the rider submits their rating, it's marked COMPLETED.

The system automatically deletes old requests 5 days after the schedule ends to keep the database clean. Every step is tracked and logged for complete transparency."

SLIDE 19: Challenges & Solutions (30 seconds)

Visual: Problems and solutions

Script: "We faced several challenges during development and solved them innovatively."

For accurate stop matching despite GPS inaccuracy, we implemented GeoHash technology with a 2.5 kilometer radius tolerance. To prevent battery drain, location updates only happen every 10 seconds and only during active trips.

For conductor verification, we require the owner's password re-authentication when registering new staff. Data cleanup happens automatically to prevent database bloat.

To handle fare disputes, everything is calculated transparently with a complete history. And for offline scenarios, we leverage Firebase's built-in offline persistence.

Every challenge was an opportunity to make Muritin more robust."

SLIDE 20: Future Enhancements (30 seconds)

Visual: Roadmap items

Script: "We have exciting plans for future versions of Muritin."

We're planning digital payment integration with bKash and Nagad, Bangladesh's most popular mobile payment platforms. Push notifications will alert users about request updates instantly.

We'll add full multi-language support with both Bengali and English. Enhanced offline mode will let the app work even without internet, syncing when connection returns.

work even without internet, syncing when connection returns.

AI-based route optimization will suggest the most efficient routes. Passenger capacity management will prevent overbooking. An emergency SOS feature will add safety, and we're exploring integration with government transit systems.

The roadmap is ambitious and user-focused."

SLIDE 21: Impact & Benefits (30 seconds)

Visual: Benefits by stakeholder

Script: "Let's talk about the real impact Muritin creates for each stakeholder.

Riders save valuable waiting time, know exactly where their bus is, get transparent pricing with no surprises, and have safe, verified boarding with OTP codes.

Conductors get efficient trip management tools, automatic digital records, performance tracking to improve their service, and easy communication with passengers.

Owners gain powerful revenue analytics, can monitor their entire fleet in real-time, manage conductors effectively, and make data-driven business decisions.

Everyone wins with Muritin. It's not just technology - it's improved lives and better business."

SLIDE 22: Technology Stack Summary (30 seconds)

Visual: Tech categories

Script: "Let me summarize our complete technology stack.

For mobile development, we use Kotlin with Jetpack Compose, Material Design 3 for beautiful UI, and Coroutines for smooth asynchronous operations.

Backend services run on Firebase Authentication, Realtime Database, with potential for Cloud Functions in the future.

Location services include the Google Maps SDK, Places API for search, Directions API for routing, and GeoFire for spatial queries.

We also use libraries like MPAndroidChart for analytics visualization, Retrofit for API calls, and Navigation Compose for seamless in-app navigation.

This modern tech stack ensures Muritin is fast, reliable, and scalable."

SLIDE 23: Demo Walkthrough (60 seconds)

Visual: Demo scenarios

Script: "Now, let me walk you through live scenarios. [SWITCH TO APP DEMO]"

First, watch as a rider books a trip - they open the app, select their current location, the system shows nearby bus stops, they choose their destination, see the fare, and submit the request. Simple and fast.

Now as a conductor, you can see pending requests, tap to accept one, and an OTP is immediately generated. The conductor can now see the rider's information and contact them.

Here's real-time tracking in action - you can see the bus moving on the map, with the pickup and destination points clearly marked. The rider watches this same view.

When the rider boards, the conductor verifies the OTP - entered correctly, and the rider is confirmed on board. The trip begins.

Finally, after reaching the destination, the conductor confirms arrival and fare collection. The rider then rates the trip - 5 stars for excellent service, adds a comment, and submits. The trip is complete.

And owners can see all of this in their analytics dashboard - trips, revenue, conductor ratings, all updated in real-time.

This is Muritin in action."

SLIDE 24: Conclusion (30 seconds)

Visual: Summary checkmarks

Script: "To conclude, Muritin is a comprehensive solution for bus service digitization.

We provide real-time tracking and communication that passengers desperately need. Multi-role support with proper access control ensures security and appropriate functionality for everyone.

The Bengali language interface makes it accessible to the local market. Our scalable architecture using Firebase means we can grow from a few buses to entire fleets across Bangladesh.

And most importantly, Muritin is ready - ready for deployment, ready for testing, ready to transform how people travel by bus.

We've built something that solves real problems with modern technology."

SLIDE 25: Questions & Answers (Remaining time)

Visual: Contact information

Script: "Thank you so much for your attention. I'm excited about Muritin's potential to revolutionize public transportation in Bangladesh






I'm now happy to answer any questions you might have about the technical implementation, the business model, user experience, scalability, or anything else about the project.

[PAUSE FOR QUESTIONS]






Thank you again, and I look forward to seeing Muritin make bus travel better for everyone!"

RECORDING TIPS:






Before Recording:

-  Practice the script 2-3 times
-  Have the app demo ready and tested
-  Close all unnecessary applications
-  Charge your devices fully
-  Test audio quality




During Recording:

-  Speak clearly and at moderate pace
-  Smile - it comes through in your voice
-  Look at the camera when not showing slides
-  Pause briefly between sections
-  If you make a mistake, pause and restart that section

Technical Setup:

-  Record at 1080p minimum (4K if possible)
-  Use external microphone if available
-  Good lighting on your face
-  Screen resolution: 1920x1080
-  Keep total under 10 minutes

Editing:

-  Cut out long pauses
-  Add subtle background music (low volume)
-  Add transition effects between slides

- 🎨 Color grade for professional look
 - 📝 Add captions if time permits
-

TIMING BREAKDOWN:

- Slides 1-5: 2 minutes 45 seconds
- Slides 6-10: 2 minutes 30 seconds
- Slides 11-15: 2 minutes 30 seconds
- Slides 16-20: 2 minutes 30 seconds
- Slides 21-23: 1 minute 30 seconds
- Slide 24: 30 seconds
- **Total: ~9 minutes 15 seconds** (leaving 45 seconds buffer)

Good luck with your presentation! 🚀