Proposal for

Wezader

Aug. 28, 2024

ISSUED BY

COMPANY NAME

REPRESENTATIVE

YOUR NAME

YOUR EMAIL

YOUR PHONE

Introduction

The courier delivery system is a software application that enables users to book courier deliveries for packages of various sizes. It provides users the ability to select a delivery type, specify package details, book pickups and deliveries, track packages, and manage their accounts. The system is designed to offer a seamless and efficient courier delivery experience for users, ensuring their packages are delivered in a timely and secure manner.

The Existing System

The existing courier delivery system relies heavily on phone-based interactions, requiring users to call in to book their deliveries. This manual process can be time-consuming and inconvenient for customers, as they have to wait on hold, provide all the necessary details verbally, and then rely on the operator to accurately record the information. The lack of a self-service online platform means that users cannot independently manage their bookings, track the status of their packages, or access their account information.

Furthermore, the phone-based nature of the existing system creates additional challenges in terms of scalability and efficiency. As the number of customers and delivery requests increases, the call center can become overwhelmed, leading to longer wait times and potential errors in the booking process. This can result in frustrated customers and delays in the overall delivery timeline, ultimately impacting the overall customer experience and the company's reputation.

Proposed System

The proposed courier delivery system aims to address the limitations of the existing phone-based system by providing a comprehensive digital platform that enhances the user experience. This mobile and web application will allow customers to independently manage their deliveries, from start to finish, without the need for constant phone interactions.

One of the key features of the proposed system is the ability for users to select the type of delivery they require, whether it be regular, express, or same-day, and input the details of their package, such as dimensions and weight. The system will then automatically generate an estimated delivery cost, enabling customers to make an informed decision about their preferred service level. This seamless selection and pricing process will offer a significant improvement over the current manual, phone-based approach.

Furthermore, the proposed system will provide users with real-time tracking capabilities, allowing them to monitor the status of their packages throughout the delivery process. Customers will also receive automated email and SMS notifications at each stage, keeping them informed and eliminating the need to continuously follow up with the courier service. The inclusion of a user-friendly account management system will further enhance the customer experience, enabling users to view their order history, reschedule deliveries, and update their profile information as needed.

Objective

The primary objective of the proposed system is to provide a seamless and efficient courier delivery experience for users. The system aims to offer the following benefits:

- Improved user experience through an intuitive and easy-to-use interface that is optimized for use across different devices.
- Enhanced functionality by offering users the ability to choose from various delivery types, estimate delivery costs, receive notifications, and manage their account information effectively.
- Increased efficiency in the delivery process through real-time package tracking and automated email/SMS notifications.
- Improved security by protecting user data through encryption and authentication.
- Increased reliability and scalability by maintaining 99.9% uptime and efficiently handling 100,000+ users.
- Enhanced integrations with payment gateways, support for multiple languages, and an admin interface for managing deliveries and tracking vans.

Functionality provided by the project

Here are the requirements for the courier delivery system, organized based on the three user roles: Admin, User, and Delivery Person.

Admin Requirements:

- Manage courier types and service offerings.
- Set pricing and delivery estimates for each courier type and item size.
- Monitor the delivery fleet and track vehicle locations.
- Manage delivery schedules and assignments
- Access real-time delivery status updates
- Generate reports on delivery metrics and performance.
- Manage user accounts and permissions
- Configure system settings and integrations

User Requirements:

- Register and manage user account.
- Select courier type (regular, express, same-day).
- Specify package details (dimensions, weight, pickup/delivery addresses).
- Receive estimated delivery cost based on selected options
- Book pickup and schedule delivery
- Reschedule or cancel deliveries
- Track the real-time status of package delivery
- Receive email and SMS notifications for delivery updates.
- Make secure online payments for deliveries
- View past delivery history and order details.

Delivery Person Requirements:

- Receive assigned delivery tasks and schedules.
- View package details and delivery addresses
- Update delivery status at each stage (picked up, in transit, delivered).
- Communicate any delivery exceptions or delays.
- Provide proof of delivery (e.g., signature, photo)
- Access turn-by-turn navigation for efficient routing
- Manage personal profile and availability
- Receive notifications and alerts related to deliveries.

Non-Functional Requirements:

- Performance: The system should process user requests within 2 seconds.
- Scalability: The system should efficiently handle 100,000+ users.
- Reliability: Maintain 99.9% uptime with high availability.
- Security: Protect user data through encryption and authentication.
- Usability: an intuitive and easy-to-use interface across devices
- The system integrates with payment gateways.
- The system supports multiple languages.
- The system is accessible via both mobile (IOS and Android) and website.
- Admin interface for managing deliveries, tracking vehicles, and monitoring system performance

Technical Feasibility

Requirement	Technology/Solution	Feasibility
Web-based application	HTML, CSS, JavaScript, React.js, and Node.js, Express,	High
Mobile Application	Flutter, react-native	High
User authentication	OAuth 2.0, JWT tokens	High
Secure payment processing	Telebirr, HelloCash, CBE Birr, Chapa	High
Real-time package tracking	GPS, geolocation APIs, WebSockets	High
Automated notifications	Email service (e.g., SendGrid), SMS service (e.g., Afro-message)	High
Database management	MySQL, PostgreSQL, MongoDB	High
Scalable infrastructure	Cloud hosting (e.g., AWS, Telecloud, Zergaw Cloud), load balancing	High
Reporting and analytics	Business intelligence tools (e.g., Tableau, Power BI)	High
Admin dashboard	React.js, Node.js, RESTful API	High
Delivery fleet management	GPS tracking, route optimization algorithms	High

Financial Feasibility

Cost Element	Estimate	Justification
Software development	250,000 - 500,000 ETB	Complexity of the system, integration of multiple components, and user-friendly interface design
Cloud hosting and infrastructure	5,000 - 20,000 ETB per month	Scaling based on user and delivery volume, data storage, and bandwidth requirements
Payment gateway integration	1,000 - 5,000 ETB per month	Percentage-based transaction fees and monthly service charges
Notification services	3000 - 10,000 ETB per month	Costs for email and SMS notifications, based on the number of users and delivery volume
Delivery fleet management	$10,000 - $50,000 per month	Costs for GPS tracking devices, fleet management software, and maintenance
Operational expenses	50,000 - 100,000 ETB per year	Customer support, maintenance, and administrative costs

The total estimated cost for the first year of implementation ranges from 500,000 to 800,000 ETB, depending on the scope, infrastructure requirements, and scale of the delivery operations. The financial feasibility of the project will depend on the projected revenue, customer acquisition rates, and the ability to achieve economies of scale over time.

Schedule Feasibility

Task	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Requirements Gathering and Analysis										
Stakeholder interviews	X	X								
Detailed requirements documentation		X	X							
Feasibility assessment		X	X							
System Design										
Architecture design		X	X							
UI/UX design (web)			X	X						
UI/UX design (mobile)				X	X					

Database design			X	X						
API design			X	X						
Development										
Front-end development (web)				X	X	X				
Front-end development (mobile)					X	X	X			
Back-end development				X	X	X	X			
Integration and testing					X	X	X	X		
Quality Assurance										
Unit testing				X	X	X	X			
Integration testing					X	X	X	X		
System testing						X	X	X		
User acceptance testing							X	X		

Deployment and Go-Live										
Staging environment setup				X						
Production deployment					X					
User training and support					X	X	X			
Maintenance and Support										
Bug fixes and patches								X	X	X
Feature enhancements									X	X
Ongoing system monitoring								X	X	X

The above table outlines the key tasks and their estimated timeline for the courier delivery system project. Here's a brief explanation of each phase:

1. **Requirements Gathering and Analysis (2 months):**
    - Conduct stakeholder interviews to gather detailed requirements for the web and mobile applications.
    - Document the functional and non-functional requirements.
    - Assess the technical and financial feasibility of the project.
2. **System Design (3 months):**
    - Design the overall system architecture, including the web and mobile components.
    - Develop the UI/UX design for both the web and mobile applications.
    - Design the database schema and the API interfaces.
3. **Development (5 months):**
    - Implement the front-end development for the web and mobile applications.
    - Develop the back-end components and integrate with the front-end.
    - Ensure seamless integration between the web and mobile applications.
4. **Quality Assurance (4 months):**
    - Perform unit testing on individual components.
    - Conduct integration testing to ensure the various components work together as expected.
    - Carry out system-level testing to validate the overall functionality and performance.
    - Involve users in the acceptance testing process to ensure the application meets their requirements.

5. **Deployment and Go-Live (2 months):**
    - Set up the staging environment to test the system in a production-like setup.
    - Deploy the applications to the production environment.
    - Provide user training and support to ensure a smooth transition.
6. **Maintenance and Support (Ongoing):**
    - Address any bug fixes and patches that are identified during the initial deployment.
    - Implement feature enhancements based on user feedback and market demands.
    - Continuously monitor the system's performance and make necessary adjustments.

The extended timeline of 10 months allows for a more thorough and iterative development process, with ample time dedicated to system design, integration, testing, and post-deployment support. This approach helps to ensure the delivery of a high-quality and well-functioning courier delivery system that meets the requirements of both the users and the organization.