ZIPIFLY-INCOGNITO PARCEL DELIVERY SYSTEM

BY

M Rafiul Faisal
ID: 201-15-3239

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Mohammad Jahangir Alam
Lecturer (Sr. Scale)
Department of CSE
Daffodil International University

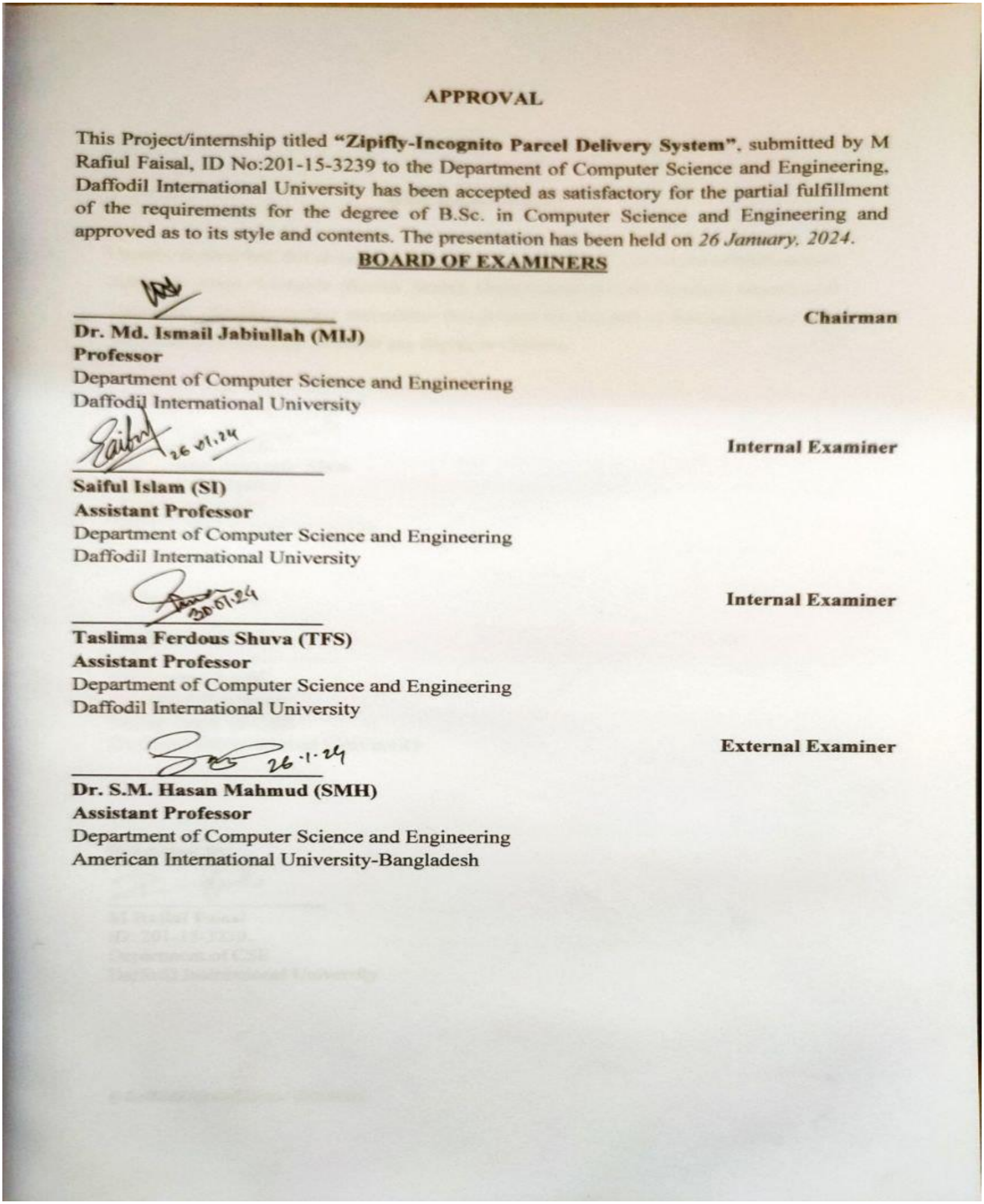
Co-Supervised By

Subhenur Latif
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2024



APPROVAL

This Project/internship titled **"Zipifly-Incognito Parcel Delivery System"**, submitted by M Rafiul Faisal, ID No:201-15-3239 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on *26 January, 2024*.

BOARD OF EXAMINERS

Chairman

Dr. Md. Ismail Jabiullah (MIJ)
Professor
Department of Computer Science and Engineering
Daffodil International University

26.01.24

Internal Examiner

Saiful Islam (SI)
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

30.01.24

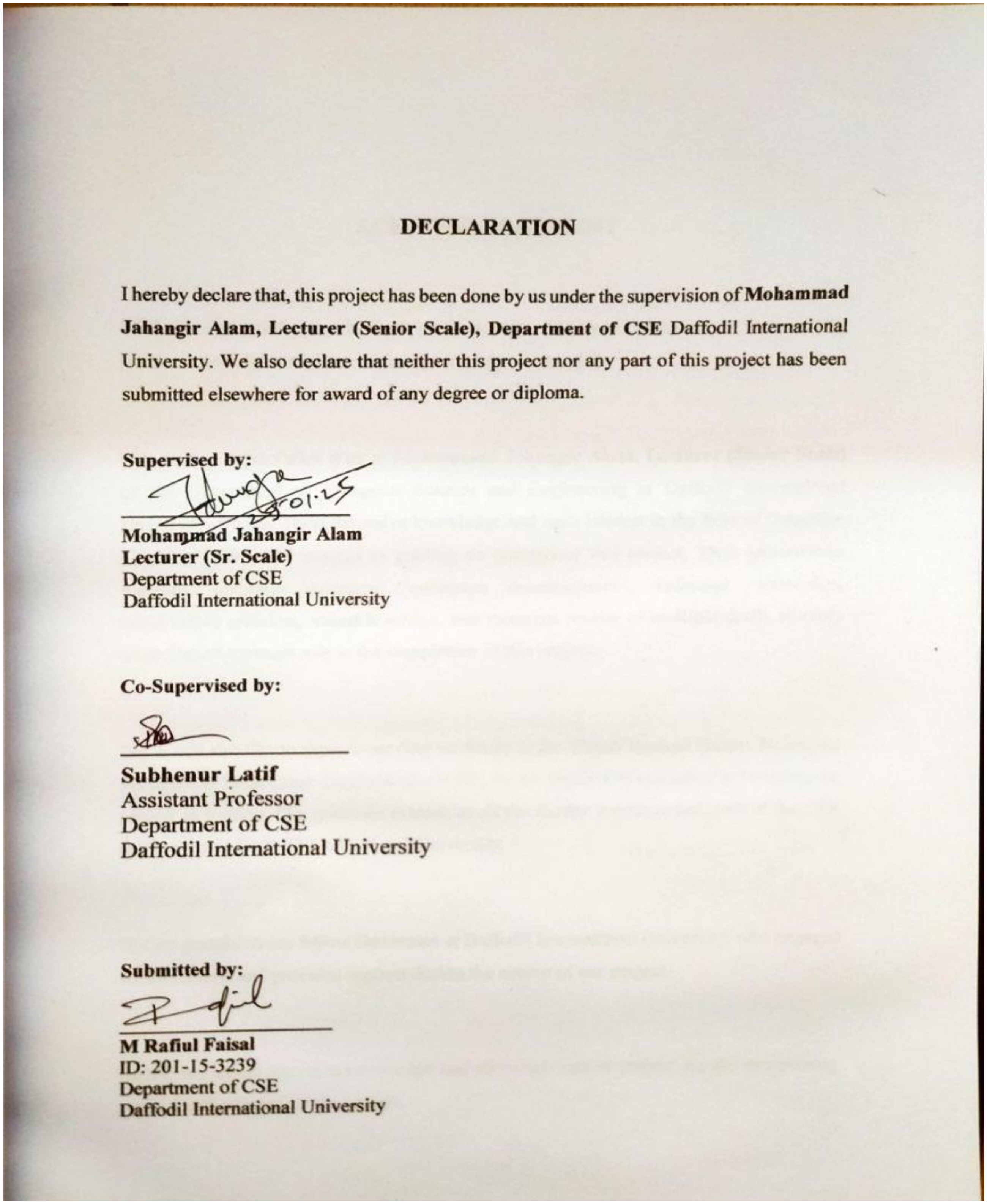
Internal Examiner

Taslima Ferdous Shuva (TFS)
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

26.1.24

External Examiner

Dr. S.M. Hasan Mahmud (SMH)
Assistant Professor
Department of Computer Science and Engineering
American International University-Bangladesh



DECLARATION

I hereby declare that, this project has been done by us under the supervision of **Mohammad Jahangir Alam, Lecturer (Senior Scale), Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

23·01·25

Mohammad Jahangir Alam
Lecturer (Sr. Scale)
Department of CSE
Daffodil International University

Co-Supervised by:

Subhenur Latif
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

M Rafiul Faisal
ID: 201-15-3239
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

We begin by extending our heartfelt gratitude to the Almighty for blessing us and enabling us to successfully complete our final year project/internship.

Our sincere appreciation goes to **Mohammad Jahangir Alam, Lecturer (Senior Scale)** of the Department of Computer Science and Engineering at Daffodil International University, Dhaka. Their extensive knowledge and keen interest in the field of "Machine Learning" were instrumental in guiding us throughout this project. Their unwavering patience, scholarly guidance, continuous encouragement, dedicated supervision, constructive criticism, valuable advice, and thorough review of multiple drafts at every stage played a crucial role in the completion of this project.

We would also like to express our deepest thanks to **Dr. Sheak Rashed Haider Noori**, the Professor & Head of the Department of CSE, for his invaluable assistance in bringing our project to fruition. Our gratitude extends to all the faculty members and staff of the CSE department at Daffodil International University.

We are grateful to our fellow classmates at Daffodil International University, who engaged in discussions and provided support during the course of our project.

Lastly, we would like to acknowledge and show our utmost respect for the unwavering support and patience of our parents.

ABSTRACT

In response to escalating privacy concerns within the parcel delivery landscape, Zipifly emerges as a groundbreaking solution—an incognito parcel delivery app that strives to redefine industry norms. The project places paramount importance on safeguarding user privacy, providing a seamless delivery experience, and addressing the pervasive issue of spam in the parcel delivery ecosystem. Motivated by a commitment to privacy and convenience, Zipifly incorporates cutting-edge technologies into its development. The frontend is crafted using ReactJS, ensuring a responsive and intuitive user interface. The backend, powered by Node Express, promises robust functionality and efficient server-side operations. MongoDB, a NoSQL database, underpins the system, offering scalability and flexibility in managing data. The user-centric design of Zipifly incorporates essential features such as onboarding, delivery tracking, quick rate checking, order history, and more. The platform empowers users to receive parcels without divulging personal information, promoting a more secure and confidential transaction process. Additionally, Zipifly introduces a unique user-code system, enabling influencers and celebrities to receive public relations gifts discreetly. This comprehensive approach not only addresses current privacy concerns but also sets Zipifly on a trajectory to establish a new industry standard. The success of this project holds the potential to reshape user expectations in the realm of parcel delivery, reducing spam and fostering a heightened sense of trust in the process.

TABLE OF CONTENS

CONTENTS **PAGE**

CHAPTER

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the contemporary landscape, privacy stands as a paramount concern for individuals, particularly noteworthy for public figures like celebrities and influencers. Despite this, parcel delivery systems often neglect these concerns, leading to the inadvertent exposure of personal information, including email addresses and phone numbers. The potential misuse of such data and the surge in spam communication underscore the pressing need for a transformative solution. In response, our innovative incognito parcel delivery app, Zipifly, emerges as a strategic and comprehensive answer to the complex challenges faced by users in the delivery ecosystem.

1.2 Motivation

The impetus behind Zipifly finds its roots in the unyielding support of a dedicated supervisor. The app's conception and development are fueled by a vision that transcends conventional paradigms—a future where privacy and convenience coalesce harmoniously. At every juncture of this journey, the motivation is sustained by the desire to set a new industry standard in the world of parcel delivery. With an eye toward empowerment, Zipifly aims to not only meet but exceed user expectations and reshape perceptions about the delivery sector.

1.3 Objectives

The primary objectives of Zipifly are manifold, encompassing a comprehensive strategy for reshaping the landscape of parcel delivery services. Paramount among these goals is the steadfast commitment to safeguarding user privacy, ensuring a secure and efficient parcel delivery experience, and delivering an app that is not only user-friendly but fortified with robust data security measures. The overarching vision is to set a new industry standard—one that fosters user trust in the handling of their data, minimizes spam, and facilitates service expansion, all while preserving privacy as an unwavering core value. In essence, Zipifly aspires to be more than an app; it is a catalyst for positive change in the dynamics of parcel delivery, championing a future where privacy is not just an expectation but a fundamental right.

1.4 Expected Outcomes

Zipifly focuses on a major outcome, that is privacy protection. But it also focuses below points:

Enhanced Privacy Protection

- Implementation of advanced encryption and security measures to ensure user data remains confidential throughout the parcel delivery process.
- Integration of incognito features, allowing users to send and receive parcels without revealing personal information.

Efficient Parcel Delivery System

- Development of a streamlined and efficient parcel delivery system that ensures timely and reliable deliveries.
- Implementation of innovative tracking mechanisms to provide real-time updates on parcel status and location.

User-Friendly Interface

- Creation of an intuitive and user-friendly app interface, promoting ease of use for individuals of varying technological proficiency.
- Incorporation of onboarding, signup, and login processes designed for seamless authentication.

Reduction of Spam and Unsolicited Communications

- Implementation of robust data security measures to prevent the unauthorized sale of customer data by businesses.

- Mitigation of spam SMS by safeguarding user contact information and preventing its misuse.

Expansion of Service with Privacy as a Core Value

- Introduction of features that allow users to receive parcels from any store without compromising their personal information.
- Establishment of a user-code system to enable celebrities and influencers to receive public relations gifts without revealing their location.

Setting a New Industry Standard

- Successful demonstration of Zipifly as a state-of-the-art incognito parcel delivery app, setting a new industry benchmark for privacy-focused delivery services.
- Recognition and adoption of Zipifly's innovative features by other parcel delivery companies, contributing to a broader shift towards prioritizing user privacy.

Empowered User Trust

- Empowerment of users to trust the delivery process, knowing that their privacy is safeguarded at every step.
- Positive user feedback and increased user satisfaction, indicating a successful integration of privacy, efficiency, and user-friendly design.

Influence on Future Innovations

- Inspiration for future developments in the parcel delivery industry, encouraging other entrepreneurs and companies to prioritize privacy and convenience in their services.
- Contributing to a paradigm shift where privacy becomes a fundamental aspect of all aspects of parcel delivery, benefiting users globally.

1.5 Project Management and Finance

In the realm of Zipifly's incognito parcel delivery app, meticulous project management and financial strategies take center stage. A judicious allocation of resources ensures optimal use of funds, while efficient project timelines are upheld through robust management methodologies. Proactive risk mitigation, scalability planning, and transparent financial reporting contribute to the project's resilience. Cost-effective development practices strike a balance between innovation and fiscal responsibility, and diversified revenue streams provide financial stability. Continuous monitoring of key performance indicators guides informed decision-making, and collaborative team efforts further enhance project efficiency. Together, these practices form a cohesive framework, ensuring the successful implementation and enduring financial sustainability of Zipifly.

1.6 Report Layout

The report unfolds systematically, commencing with an introduction to the Zipifly project. Subsequent chapters include background study, requirement specifications, design specifications, implementation and testing details, societal and environmental impacts, and a conclusion with future prospects. The reference section acknowledges sources contributing to the project's development. This structured layout offers a comprehensive insight into the Zipifly Incognito Parcel Delivery App.

CHAPTER 2

BACKGROUND STUDY

2.1 Preliminaries/Terminologies

In understanding the landscape of the Zipifly Incognito Parcel Delivery App, certain key terms and concepts warrant clarification. Preliminaries and terminologies provide a foundation for comprehending the intricate details of the project, offering readers a shared understanding of crucial elements integral to the discussion.

2.2 Related Work

While various delivery system apps saturate the market, the critical aspect of ensuring maximum privacy for users is notably absent in existing solutions. Although the notion of concealing personal information may be novel, it is imperative to acknowledge the existence of other parcel delivery systems that have paved the way for the conceptualization and development of our innovative Zipifly incognito parcel delivery app. These existing systems have offered valuable insights and lessons, providing a foundation upon which Zipifly seeks to revolutionize the privacy landscape in parcel delivery.
Available app: Pathao, RedX, Delivery Tiger etc.

2.3 Comparative Analysis

A comparative analysis serves as a crucial step in discerning the unique features and advantages that Zipifly brings to the parcel delivery landscape. By comparing its attributes to existing delivery apps such as Pathao and RedX, the chapter aims to highlight the differentiating factors and innovations that set Zipifly apart. This analysis sets the stage for understanding how Zipifly addresses the privacy concerns and challenges prevalent in the current market.

2.4 Scope of the Problem

The scope of the problem encapsulates the overarching challenges faced by users in the existing parcel delivery systems. From privacy concerns to the compromises made with personal information, this section elucidates the multifaceted aspects that Zipifly seeks to address. The scope of the problem serves as a foundational understanding, laying the groundwork for the subsequent exploration of solutions.

2.5 Challenges

The development of the Zipifly Incognito Parcel Delivery App presents several challenges that require careful consideration for the successful implementation of the project:

Privacy Preservation Algorithms

Crafting algorithms that ensure optimal privacy preservation during parcel delivery without compromising efficiency demands intricate solutions. Striking a balance between incognito features and delivery speed poses a challenge.

User Adoption and Trust

Encouraging users, including parcel senders and recipients, to adopt a new system and trust in the incognito features necessitates targeted education and support. Convincing users to adapt to changes in parcel delivery processes is vital for the app's success.

Data Security and Encryption

Establishing robust data security measures and encryption protocols to safeguard sensitive user information stored on the cloud-based server is imperative. Maintaining trust and compliance with privacy regulations is a critical challenge.

Integration with Delivery Ecosystem

Seamlessly integrating Zipifly with existing delivery systems, logistics platforms, and e-commerce frameworks poses a challenge. Compatibility with diverse systems and ensuring a smooth transition for businesses and users is a key consideration.

Tailored Customization for Businesses

Designing a system that offers customization options to meet the specific needs of diverse businesses while maintaining a consistent and user-friendly experience is a challenge. The app must be adaptable to various business models and requirements.

Scalability in Parcel Volume

Ensuring that the app can efficiently handle a growing number of users, increasing parcel volumes, and expanding data requirements as businesses scale up is a crucial challenge for long-term success.

Cross-Platform Compatibility

Developing a user-friendly app that is compatible with various devices and operating systems is essential. Achieving a seamless and consistent experience across different platforms is a challenge that demands attention to diverse technological environments.

Addressing these challenges is paramount for the Zipifly Incognito Parcel Delivery App to successfully deliver a revolutionary, secure, and user-friendly solution to the contemporary concerns in the parcel delivery domain.

CHAPTER 3

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Business Process Modeling

Business Process Modeling (BPM) is a methodical representation of business processes to understand, analyze, and optimize the flow of activities within an organization. It involves creating visual representations that depict how various tasks, activities, and data are interconnected and flow through different stages to achieve specific business objectives.

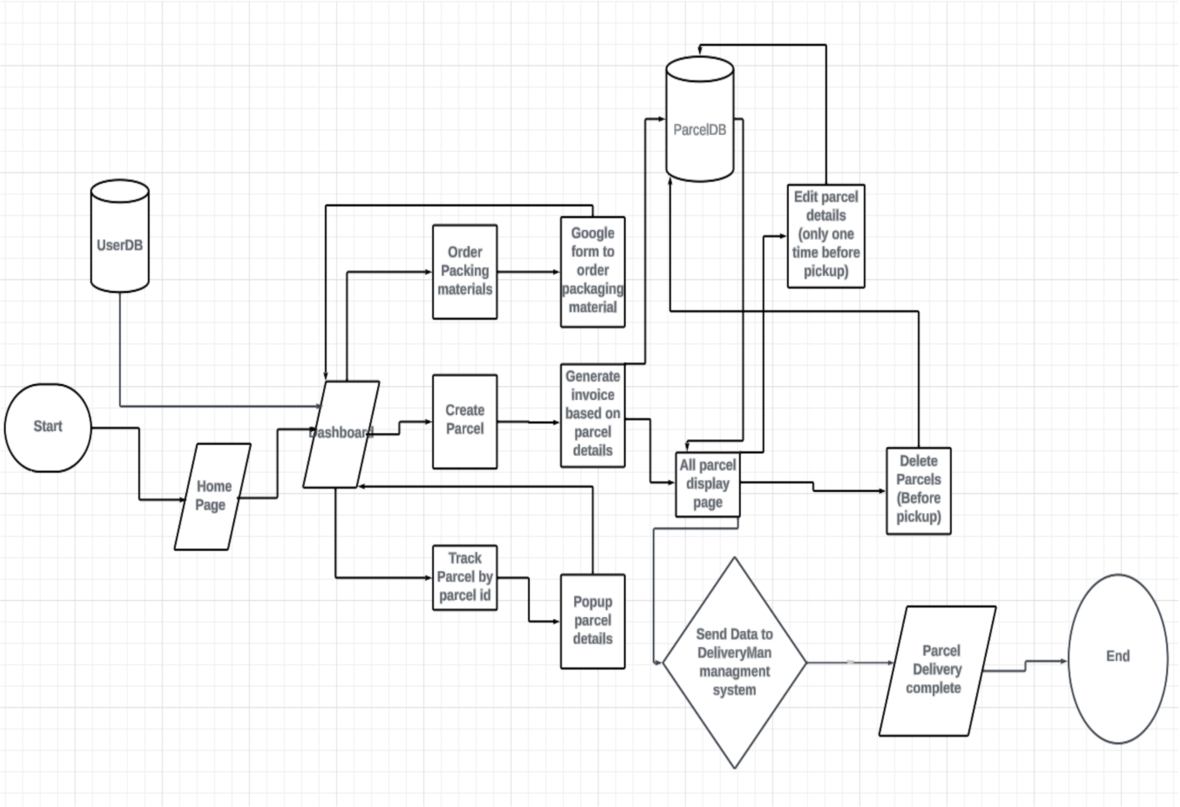

ParcelDB
UserDB
Order Packing materials
Google form to order packaging material
Edit parcel details (only one time before pickup)
Start
Home Page
Dashboard
Create Parcel
Generate invoice based on parcel details
All parcel display page
Delete Parcels (Before pickup)
Track Parcel by parcel id
Popup parcel details
Send Data to DeliveryMan managment system
Parcel Delivery complete
End

Figure 3.1.1: Business Process Model

3.2 Requirement Collection and Analysis

Requirement collection and analysis form a pivotal phase in the development of the Zipifly Incognito Parcel Delivery App, involving a meticulous evaluation of stakeholder needs, expectations, and constraints. This process ensures the effective design of the system to meet the identified requirements. The following provides a concise overview:

Useful Prerequisites

Key features and capabilities essential to meeting stakeholder needs and expectations in the parcel delivery domain:

- **Incognito Features:** Anonymized parcel sending and receiving to safeguard user privacy.
- **Efficient Tracking:** Real-time tracking for users to monitor the status and location of their parcels.
- **User Authentication:** Secure onboarding, signup, and login processes for reliable user identification.
- **Seamless Integration:** Compatibility with various devices and platforms for a versatile user experience.

Non-Functional Prerequisites

System characteristics and constraints shaping user experience and performance in the parcel delivery app:

- **Usability:** An intuitive and user-friendly interface ensuring ease of navigation.
- **Scalability:** Ability to handle increasing user numbers, parcel volumes, and data without compromising performance.
- **Security Measures:** Robust data security and encryption to protect sensitive user information.

- **Cross-Platform Compatibility:** Development of a web application compatible with diverse devices and browsers for a consistent user experience.

By comprehensively addressing these prerequisites, the Zipifly Incognito Parcel Delivery App is poised to meet user expectations, ensure data security, and deliver an efficient and user-friendly solution to contemporary challenges in the parcel delivery sector.

Functional Requirement Prerequisites:

The function requirements for the Zipifly Incognito Parcel Delivery App encapsulate essential features such as incognito parcel sending, real-time tracking, and secure user authentication etc. These functionalities ensure a seamless and privacy-focused parcel delivery experience for users, forming the core foundation of the application. Referencing the detailed table will provide an in-depth understanding of each function's specifications and interactions.

Table 3.2.1: Functional requirements table

FR01	Authentication (login and registration)
Description	Admin, customers and deliveryman have to login otherwise you cannot use the system. Only registered customer can login into the system.
stakeholder	Admin, Customer, Deliveryman
FR02	Unique user-code
Description	Every user will get unique shareable user-code.
Stakeholder	Customers
FR03	Add location
Description	User can add multiple location for their account.
Stakeholder	Customers
FR04	Pickup request

Description	Customer(sender) can make a pickup request.
Stakeholder	Customers (sender)
FR05	Drop off
Description	Customers can see his/her of all current delivery and its status.
Stakeholder	Customers
FR06	Check Rates
Description	User can check rates by putting information about the delivery and product.
Stakeholder	Customers
FR07	History
Description	User can see his/her all previous delivery history.
Stakeholder	Customers
FR08	View tracking details
Description	By clicking on a delivery card, user can track and see details of delivery.
Stakeholder	Customers, Admin
FR09	Check Offers
Description	Customers can see current discount and offer.
Stakeholder	Customers
FR10	Online wallet
Description	Customers can see his/her receivable amount from online wallet.
Stakeholder	Customers
FR11	Receive pickup request
Description	Deliveryman accept the request and pick up the parcel and deliver to store.

Stakeholder	Deliveryman
FR12	Receive delivery request
Description	Deliveryman accept the delivery request and deliver to receiver.
Stakeholder	Deliveryman

3.3 Use Case Modeling and Description

Illustrates the interactions between users and the Zipifly Incognito Parcel Delivery App, showcasing various scenarios where users engage with key features, ensuring a comprehensive understanding of system functionality.

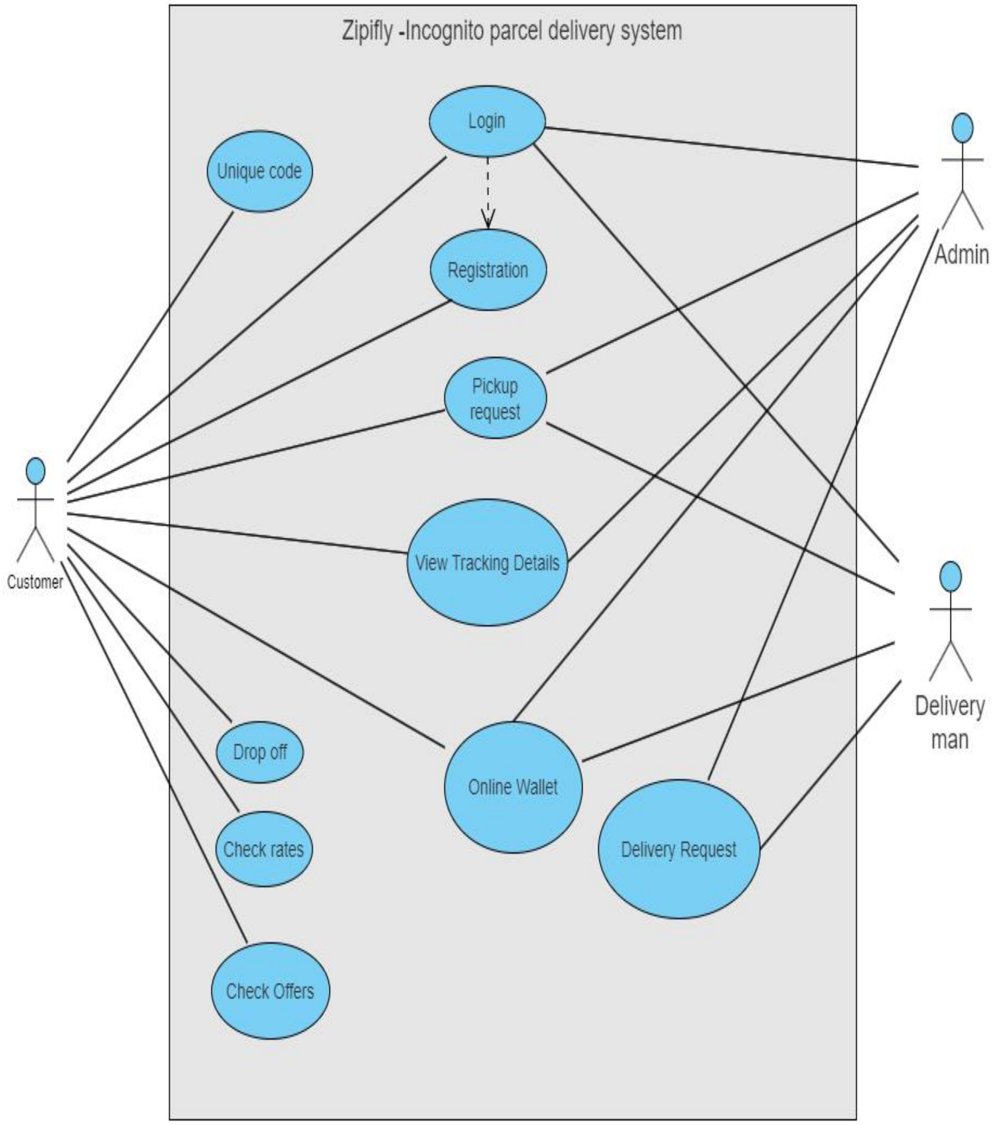


Figure 3.3.1: Use case diagram of Zipifly-incognito parcel delivery system.

Table 3.3.1: Use case description login

<table>
<tr><td>Use Case 01</td><td>Login</td></tr>
<tr><td>Goal</td><td>Customer login is must for accessing the system.
Admin and Deliveryman login is must for accessing the system.</td></tr>
<tr><td>Preconditions</td><td>Need to username and password</td></tr>
<tr><td>Success-end condition</td><td>Come into the system and access the dashboard</td></tr>
<tr><td>Failed-end condition</td><td>Can’t view the dashboard.</td></tr>
<tr><td>Actors</td><td>• Admin.
• Customers
• Delivery man</td></tr>
<tr><td>Trigger</td><td>Login into the system.</td></tr>
<tr><td>Description</td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Come to login page</td></tr>
<tr><td>2</td><td>Put username and password</td></tr>
<tr><td>3</td><td>Push login button</td></tr>
</table></td></tr>
</table>

Table 3.3.2: Use case description registration

<table>
<tr><td>Use Case 02</td><td>Registration</td></tr>
<tr><td>Goal</td><td>Customer need to register into the system to login.</td></tr>
<tr><td>Preconditions</td><td>N/A</td></tr>
<tr><td>Success-end condition</td><td>Success popup and redirect to login page for login into the system.</td></tr>
<tr><td>Failed-end condition</td><td>Failure popup.</td></tr>
<tr><td>Actors</td><td>• Customers</td></tr>
<tr><td>Trigger</td><td>Registration into the system.</td></tr>
<tr><td>Description</td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Come to registration page</td></tr>
<tr><td>2</td><td>Put all the information</td></tr>
<tr><td>3</td><td>Push signup button</td></tr>
</table></td></tr>
</table>

Table 3.3.3: Use case description username

<table>
<tr><td>Use Case 03</td><td>Unique username</td></tr>
<tr><td>Goal</td><td>Customer will set a unique user-name</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>Profile will have an unique user-name.</td></tr>
<tr><td>Failed-end condition</td><td>A warning popup</td></tr>
<tr><td>Actors</td><td>• Customers</td></tr>
<tr><td>Trigger</td><td>User-code for customers</td></tr>
<tr><td>Description</td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Come to profile page</td></tr>
<tr><td>2</td><td>Click on unique user-code option</td></tr>
<tr><td>3</td><td>Write unique user-name</td></tr>
<tr><td>4</td><td>Press set button</td></tr>
</table></td></tr>
</table>

Table 3.3.4: Use case description pickup request

<table>
<tr><td>Use Case 04</td><td>Pickup request from customer</td></tr>
<tr><td>Goal</td><td>Customer(sender) make a parcel pickup request</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>A successfully parcel pickup request and redirected to dashboard.</td></tr>
<tr><td>Failed-end condition</td><td>A warning popup and redirected to dashboard.</td></tr>
<tr><td>Actors</td><td>• Customers (sender)</td></tr>
<tr><td>Trigger</td><td>Pickup request to the system.</td></tr>
<tr><td>Description</td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Come to pick up request page</td></tr>
<tr><td>2</td><td>Fill up necessary information about pickup</td></tr>
<tr><td>3</td><td>Press pickup button</td></tr>
</table></td></tr>
</table>

Table 3.3.5: Use case description track parcel

<table>
<tr><td>Use Case 05</td><td>View/track parcel details</td></tr>
<tr><td>Goal</td><td>User to see details about his/her parcel.</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>View parcel delivery details component.</td></tr>
<tr><td>Failed-end condition</td><td>N/A</td></tr>
<tr><td>Actors</td><td>• Customers</td></tr>
<tr><td>Trigger</td><td>View parcel details</td></tr>
<tr><td>Description</td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Click on parcel components</td></tr><tr><td>2</td><td>Parcel details page shown</td></tr></table></td></tr>
</table>

Table 3.3.6: Use case description online wallet

<table>
<tr><td>Use Case 06</td><td>Online Wallet</td></tr>
<tr><td>Goal</td><td>User can track his receivable amount.</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>Details of receivable amount</td></tr>
<tr><td>Failed-end condition</td><td>Error message</td></tr>
<tr><td>Actors</td><td>• Customers</td></tr>
<tr><td>Trigger</td><td>Online wallet page</td></tr>
<tr><td>Description</td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Click on online wallet icon</td></tr><tr><td>2</td><td>Come to online wallet page</td></tr></table></td></tr>
</table>

Table 3.3.7: Use case description delivery request

<table>
<tr><td>Use Case 07</td><td>Delivery Request</td></tr>
<tr><td>Goal</td><td>Admin pass the pickup request as delivery request to a deliveryman</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>Deliveryman got details about the delivery.</td></tr>
<tr><td>Failed-end condition</td><td>Error message.</td></tr>
<tr><td>Actors</td><td>• Admin
• Deliveryman</td></tr>
<tr><td>Trigger</td><td>Online wallet page</td></tr>
<tr><td>Description</td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Admin makes a delivery request to the system.</td></tr>
<tr><td>2</td><td>Deliveryman accept the delivery request.</td></tr>
<tr><td>3</td><td>Delivery details visible to deliveryman</td></tr>
</table></td></tr>
</table>

Table 3.3.8: Use case description check rates

<table>
<tr><td>Use Case 08</td><td>Check Rates</td></tr>
<tr><td>Goal</td><td>User can check the rates for particular address and items</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>Details of receivable amount</td></tr>
<tr><td>Failed-end condition</td><td>Error message popup</td></tr>
<tr><td>Actors</td><td>• Customers</td></tr>
<tr><td>Trigger</td><td>View rates for given information.</td></tr>
<tr><td>Description</td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Click on check rates icon</td></tr>
<tr><td>2</td><td>Come to check rates page</td></tr>
<tr><td>3</td><td>Put necessary information.</td></tr>
<tr><td>4</td><td>Display calculated rates as popup.</td></tr>
</table></td></tr>
</table>

Table 3.3.9: Use case description check offers

<table>
<tr><td>Use Case 09</td><td>Check Offers</td></tr>
<tr><td>Goal</td><td>User can see available offers.</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>All currently ongoing offers.</td></tr>
<tr><td>Failed-end condition</td><td>No offers message</td></tr>
<tr><td>Actors</td><td>• Customers</td></tr>
<tr><td>Trigger</td><td>View offers page</td></tr>
<tr><td>Description</td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Click on check offers icon.</td></tr><tr><td>2</td><td>Come to check offer page.</td></tr></table></td></tr>
</table>

Table 3.3.10: Use case description drop off

<table>
<tr><td>Use Case 10</td><td>Drop off</td></tr>
<tr><td>Goal</td><td>User to see details about his/her all parcel.</td></tr>
<tr><td>Preconditions</td><td>Need to be login.</td></tr>
<tr><td>Success-end condition</td><td>View all soon to delivered parcel list.</td></tr>
<tr><td>Failed-end condition</td><td>N/A</td></tr>
<tr><td>Actors</td><td>• Customers</td></tr>
<tr><td>Trigger</td><td>View delivery parcels list</td></tr>
<tr><td>Description</td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Click on drop off button</td></tr><tr><td>2</td><td>Come to parcel delivery list page</td></tr></table></td></tr>
</table>

3.4 Logical Data Model

3.4.1 Entity Relation Model

Table 3.4.1.1: Entity Relation table

User	(uid, username, fullname, email, location, division, contract_number, parcel_sent, parcel_received)
Parcel	(pid, sender, receiver, status, weight, category, cash_collection, total_cash)
Order	(orid, pid, sender, receiver, delivery_man, pickup_date, delivery_date, status,)
Delivery man	(dmid, verification_number, fullname, orders)
Admin / Super Admin	(aid, role, fullname, employee_id)

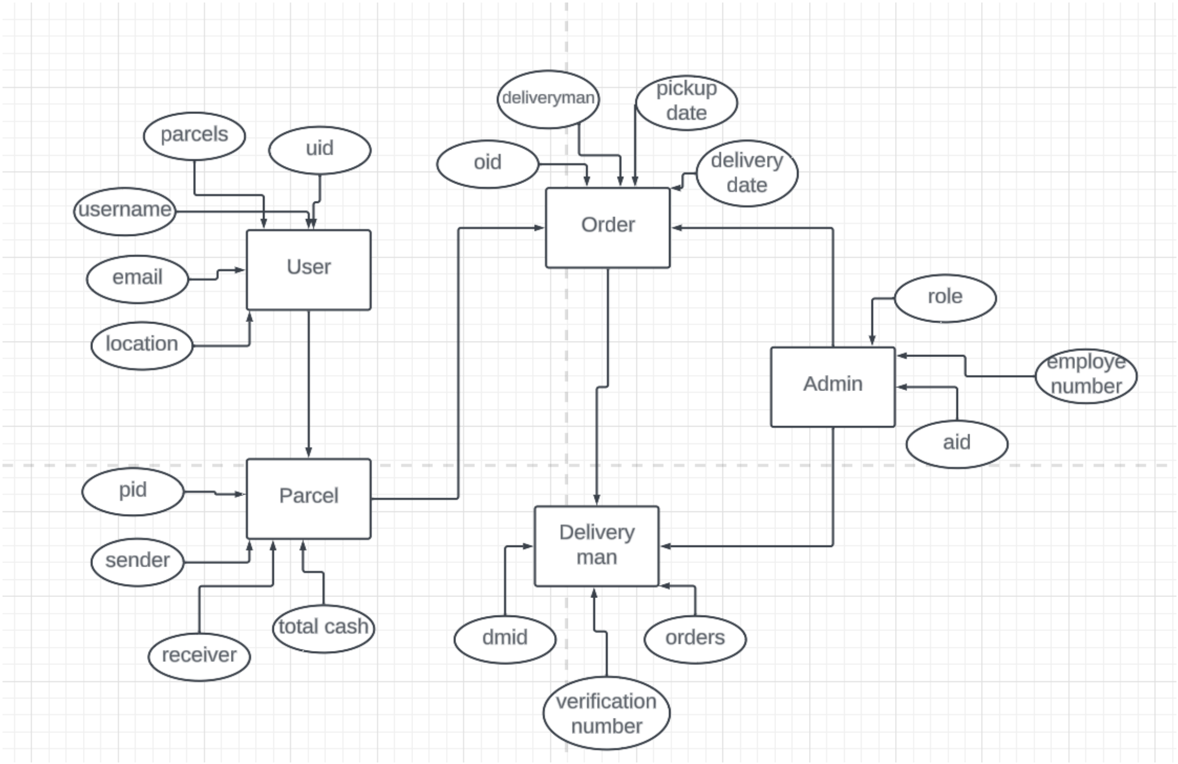


Figure 3.4.1.1: Entity Relation Diagram

3.4.2 Logical Data Model Diagram

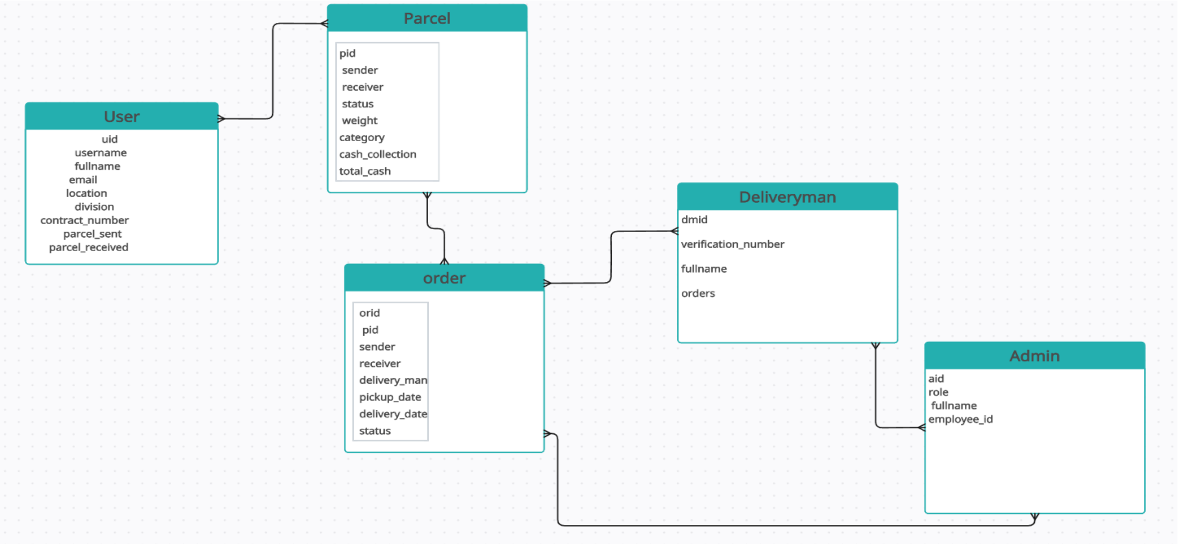


Figure 3.4.2.1: Logical Data Model Diagram

3.5 Design Requirements

The design requirements for the Zipifly Incognito Parcel Delivery App encompass both the front-end and back-end aspects, ensuring a cohesive and secure application. The integration of cutting-edge technologies plays a pivotal role in achieving the project's objectives.

Frontend: React JS, Material UI, HTML, CSS.
Backend: MongoDB, Node JS, Express JS
Security: JSON web token, bcrypt hash function.

By adhering to these design requirements, Zipifly aims to deliver a secure, user-friendly, and innovative incognito parcel delivery app that addresses contemporary challenges and sets new standards in the delivery domain.

CHAPTER 4

DESIGN SPECIFICATION

4.1 Front-end Design

The front-end of our web application, representing the direct interface for user interaction, has been meticulously crafted for optimal engagement and accessibility. Employing React as the primary framework and Material-UI for design components, we've established a user-friendly interface that ensures seamless interactions. The foundation of our front-end design integrates HTML for content creation, CSS for styling, and Formik for form validation. Additionally, JavaScript is employed to enhance the user experience with the integration of small, efficient applications. The amalgamation of these technologies forms a robust and visually appealing front-end for our project.

HTML: Hyper Text Markup Language

Websites frequently utilize HTML, a widely adopted markup language, for their creation. HTML is employed to structure and present various elements, including tables and information that require a structured format. This markup language is instrumental in ensuring that content on our websites, particularly data like tables, is appropriately organized and can be presented in a styled and visually appealing manner rather than remaining as plain text.

CSS: Cascading Style Sheet

Divergent from HTML, CSS, or Cascading Style Sheet, serves as a styling language employed to define the presentation aspects of an HTML document. In our project, CSS plays a pivotal role in dictating the appearance of various elements, including but not limited to layout, font size, text alignment, font family, background images, borders, and more. It ensures a cohesive and visually appealing design across every facet of our application.

JavaScript

JavaScript functions as a scripting language, enabling the creation of miniature applications within the broader scope of our project. This scripting language is instrumental in crafting intricate interface components, adding a layer of interactivity to our application. Positioned as the third tier in the conventional web technology stack, JavaScript enhances the user experience by introducing dynamic and responsive elements to the overall design.

ReactJS: JavaScript Library

ReactJS, a powerful JavaScript library, distinguishes itself by facilitating the development of dynamic user interfaces. It operates as the core framework for our project, offering a component-based structure that enhances modularity and reusability. ReactJS empowers us to efficiently manage UI states and seamlessly update the interface in response to user interactions. In our project, ReactJS plays a central role in crafting a responsive and interactive user experience.

Material-UI

Material-UI complements ReactJS in our front-end design, serving as a robust React component library with pre-designed elements following the Material Design principles. As a set of React components, Material-UI streamlines the creation of visually consistent and aesthetically pleasing user interfaces. It provides a plethora of customizable components such as buttons, cards, and navigation bars, contributing to the overall cohesiveness and professional appearance of our application.

4.2 Back-end Design

In our back-end development, we rely on a powerful tech stack. Node.js forms the core, enabling efficient server-side JavaScript execution with its non-blocking, event-driven architecture. Express.js complements Node.js, simplifying the creation of robust web applications and facilitating RESTful API development. The persistence layer is driven by MongoDB, a flexible NoSQL database seamlessly integrated with Node.js and Express.js.

Together, these technologies create a scalable, responsive, and efficient server-side foundation for our project.

Node.js: JavaScript Runtime Environment

Node.js, functioning as a JavaScript runtime environment, forms the backbone of our back-end infrastructure. It enables server-side JavaScript execution, allowing for efficient handling of requests and responses. Node.js fosters a non-blocking, event-driven architecture, contributing to the scalability and responsiveness of our application's server-side operations.

Express.js: Web Application Framework for Node.js

Express.js, a minimalist web application framework for Node.js, further enhances our back-end development. It simplifies the creation of robust and scalable web applications by providing a set of essential features and middleware. Express.js facilitates the routing of requests, handling HTTP methods, and streamlining the development of RESTful APIs. Its lightweight yet powerful nature aligns seamlessly with our project's goals, offering a solid foundation for building efficient server-side components.

MongoDB: NoSQL Database

MongoDB serves as our project's NoSQL database, offering a flexible and scalable solution for data storage. Its document-oriented structure allows for the efficient organization and retrieval of data. MongoDB seamlessly integrates with Node.js and Express.js, fostering a cohesive interaction between the server and the database. This NoSQL database solution plays a crucial role in managing and storing data with agility, supporting the dynamic requirements of our application.

JWT (JSON Web Tokens)

Utilizing JWT as our authentication mechanism ensures not only the secure transmission of information between the client and server but also the validation of the authenticity of each request. This token-based approach enhances security by securely transmitting user credentials, playing a pivotal role in preventing unauthorized access and maintaining the integrity of user sessions.

bcrypt Hashing Function

The storage and management of user passwords are fortified through the implementation of the bcrypt hashing function. This cryptographic hash function adds an extra layer of security by securely transforming user passwords into irreversible hashes. Even in the event of a data breach, bcrypt ensures that stored passwords remain resistant to decryption, thus safeguarding user credentials effectively.

HTTPS Implementation

Strictly enforce HTTPS to encrypt data in transit, ensuring the confidentiality and integrity of information exchanged between the client and server.

Input Validation

Implement stringent input validation at both client and server levels to prevent common security threats, such as SQL injection and cross-site scripting (XSS) attacks. This fortifies our defense against malicious input.

Rate Limiting

Employ rate limiting mechanisms on API endpoints to thwart abuse and mitigate the risk of brute force attacks. This proactive measure ensures that repetitive, automated requests are effectively controlled.

Session Management

Strengthen session management practices by incorporating best practices such as token expiration, session token regeneration, and secure storage of session data. This enhances the overall security of user sessions.

4.3 Interaction Design and User Experience (UX)

In the context of our project, User Experience (UX) design is paramount, shaping the overall interaction users have with our application. The central objective of UX design is to craft experiences that are not only intuitive and efficient but also enjoyable, aligning seamlessly with users' needs and expectations. Every facet of user interaction is considered, spanning visual design, layout, navigation, and overall usability. Our commitment to a positive UX ensures that users can effortlessly and delightfully engage with our application, fostering satisfaction and loyalty.

4.4 Implementation Requirements

We need software for building and controlling the project.

1. Visual Studio Code (for writing codes).

2. MongoDB Atlas (for online database solution).

3. Postman (for API testing)

4. Figma (for UI development)

CHAPTER 5

IMPLEMENTATION AND TESTING

The implementation phase showcases the seamless integration of technologies, while testing ensures the reliability, functionality, and security of our incognito parcel delivery app.

5.1 Implementation of Database

In the implementation of the Zipifly Incognito Parcel Delivery App's database, MongoDB serves as the foundation, providing a flexible and scalable solution to manage crucial data. The database consists of four primary collections: User, Parcel, Order, and Delivery Man.

User Collection (users)

- uid: Unique identifier for each user.
- username: User's chosen username.
- fullname: Full name of the user.
- email: Email address for communication.
- location: User's current location.
- division: Geographical division information.
- contract_number: Contact number for communication.
- parcel_sent: Counter for parcels sent.
- parcel_received: Counter for parcels received.

Parcel Collection (parcels)

- pid: Unique identifier for each parcel.
- sender: Reference to the User who sent the parcel.
- receiver: Reference to the User who will receive the parcel.
- status: Current status of the parcel.
- weight: Weight of the parcel.

- category: Categorization of the parcel.
- cash_collection: Amount to be collected on delivery.
- total_cash: Total cash amount associated with the parcel.

Order Collection (orders)

- orid: Unique identifier for each order.
- pid: Reference to the Parcel associated with the order.
- sender: Reference to the User who sent the parcel.
- receiver: Reference to the User who will receive the parcel.
- delivery_man: Reference to the assigned Delivery Man.
- pickup_date: Date of parcel pickup.
- delivery_date: Expected delivery date.
- status: Current status of the order.

Delivery Man Collection (deliverymen)

- dmid: Unique identifier for each delivery man.
- verification_number: Verification number for identification.
- fullname: Full name of the delivery man.
- orders: List of orders assigned to the delivery man.

Admin / Super Admin Collection (admins)

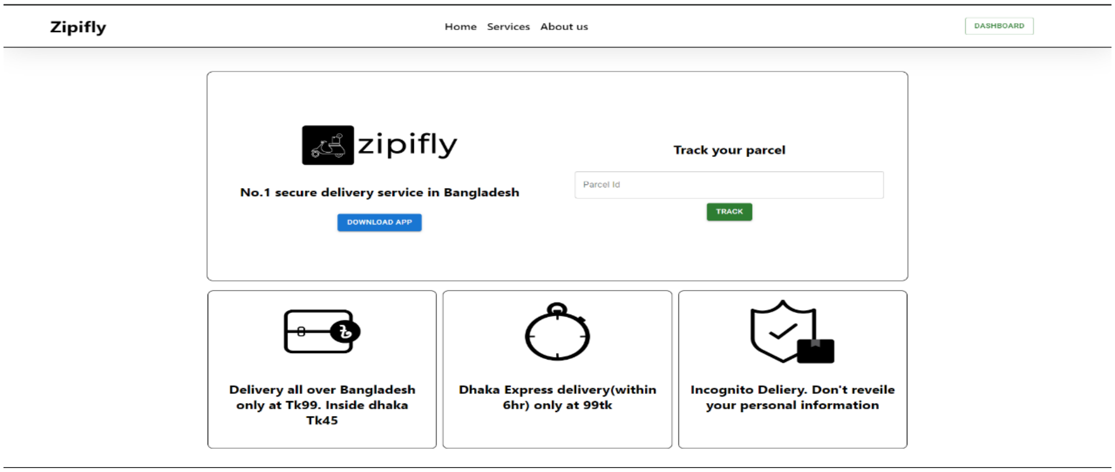
- aid: Unique identifier for each admin.
- role: Designation indicating admin or super admin.
- fullname: Full name of the admin.
- employee_id: Unique identifier for employee tracking.

This MongoDB implementation ensures efficient data organization and retrieval, supporting the dynamic requirements of the Zipifly Incognito Parcel Delivery App. The structured collections establish relationships, facilitating seamless interactions and operations within the application.

5.2 Implementation of Front-end Design

Utilizing ReactJS and Material-UI, we implement a visually appealing and intuitive frontend. Features like onboarding, signup, login, order tracking, and rate checking are developed to ensure a seamless, user-centric design. This frontend implementation breathes life into our incognito parcel delivery app, establishing a foundation for a responsive, engaging, and privacy-focused user interface.

5.2.1 Homepage



No.1 secure delivery service in Bangladesh

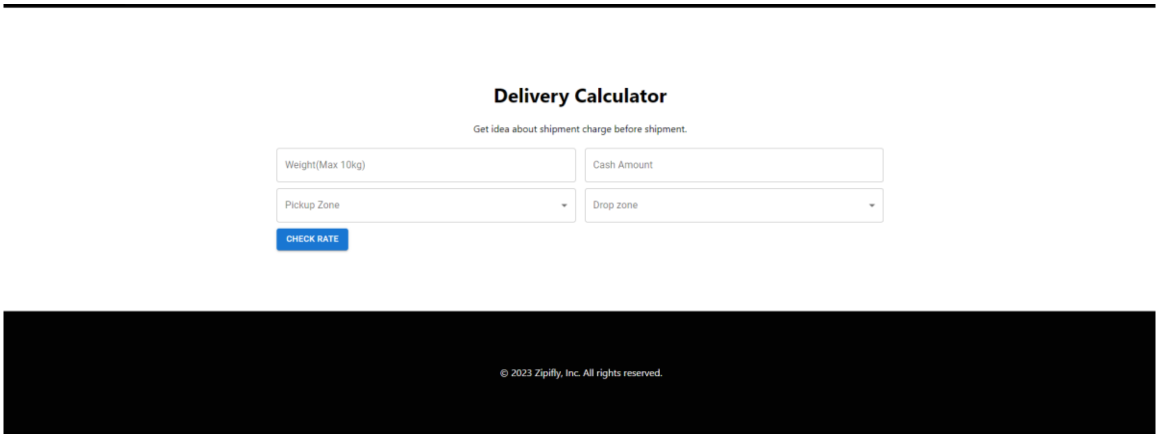
Delivery all over Bangladesh only at Tk99. Inside dhaka Tk45

Dhaka Express delivery(within 6hr) only at 99tk

Incognito Deliery. Don't reveile your personal information

Figure 5.2.1: Home page

5.2.2 Check Rates

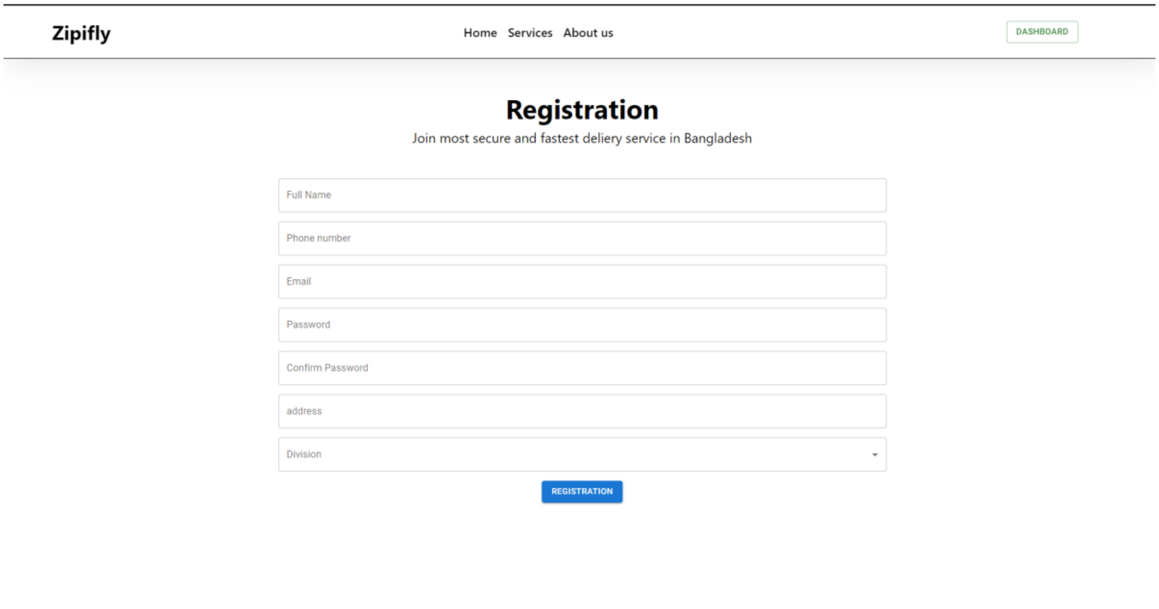


Get idea about shipment charge before shipment.

© 2023 Zipifly, Inc. All rights reserved.

Figure 5.2.2: Check Rates

5.2.3 Registration

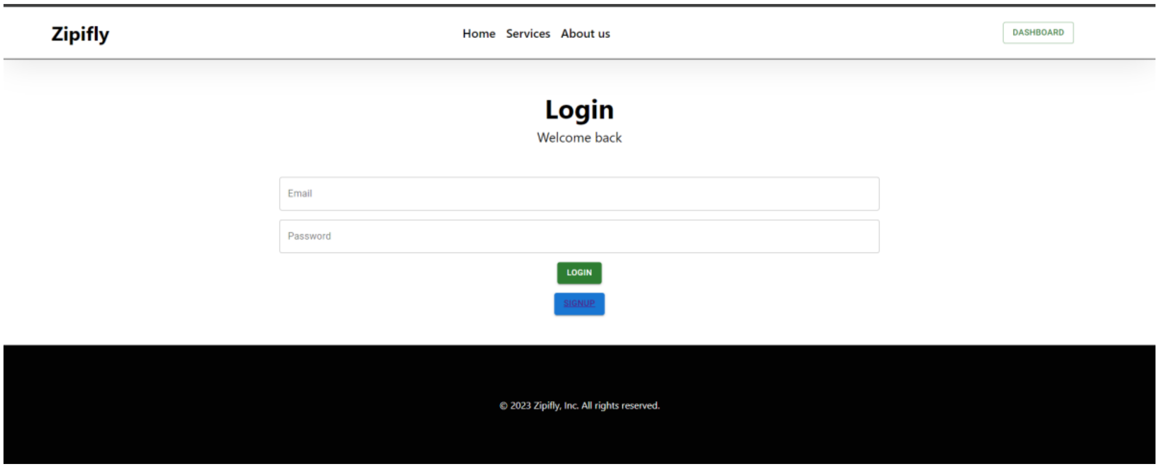


Registration

Join most secure and fastest deliery service in Bangladesh

Figure 5.2.3: Registration

5.2.4 Login



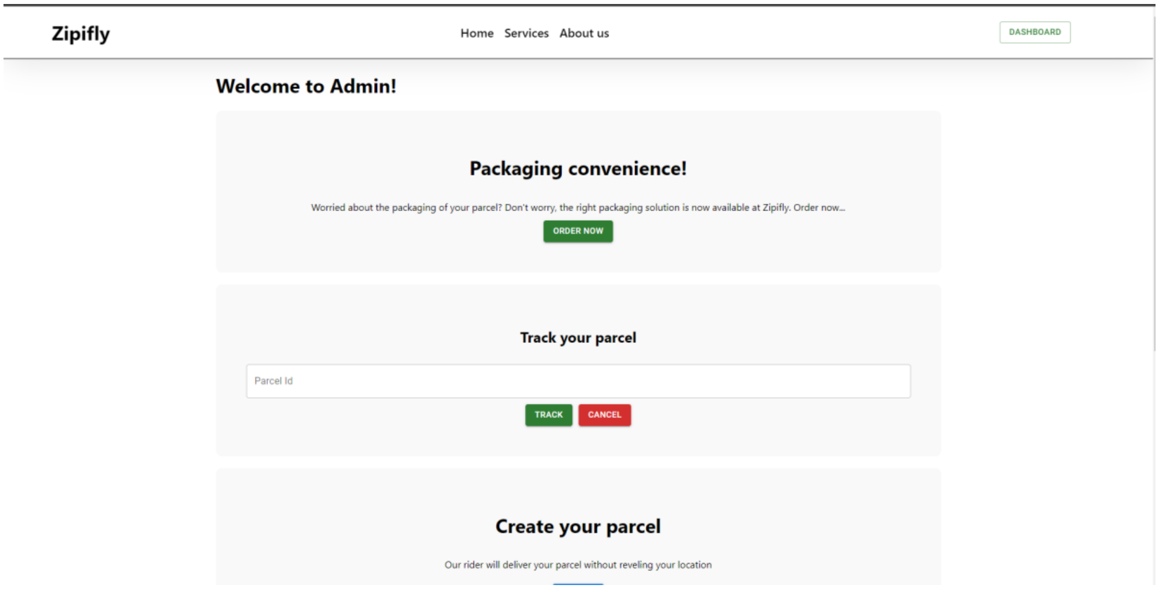
Login

Welcome back

© 2023 Zipifly, Inc. All rights reserved.

Figure 5.2.4: Login

5.2.5 Dashboard



Welcome to Admin!

Packaging convenience!

Worried about the packaging of your parcel? Don't worry, the right packaging solution is now available at Zipifly. Order now...

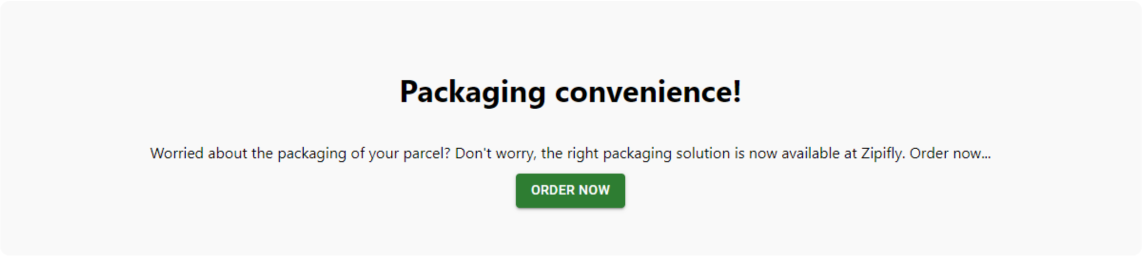
Track your parcel

Create your parcel

Our rider will deliver your parcel without reveling your location

Figure 5.2.5: Dashboard

5.2.6 Order Packaging Material



Packaging convenience!

Worried about the packaging of your parcel? Don't worry, the right packaging solution is now available at Zipifly. Order now...

Figure 5.2.6: Order Packaging Materials

5.2.7 Track Parcel

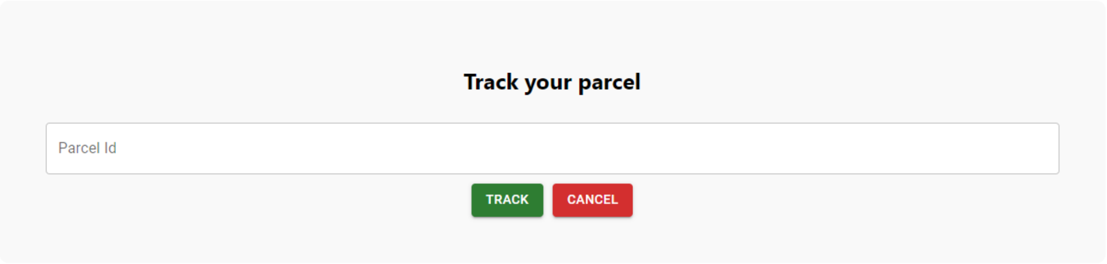
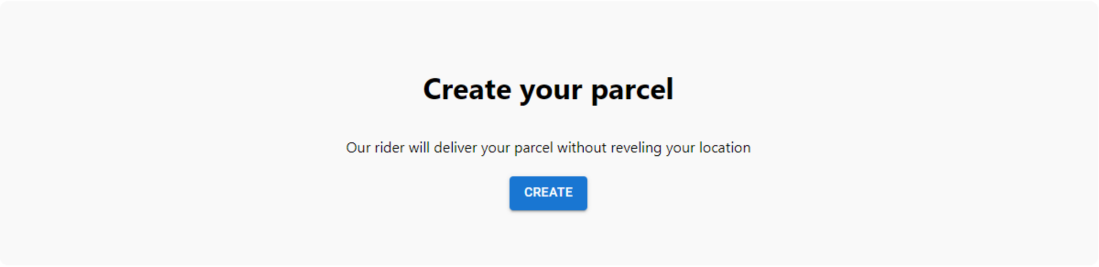


Figure 5.2.7: Track parcel

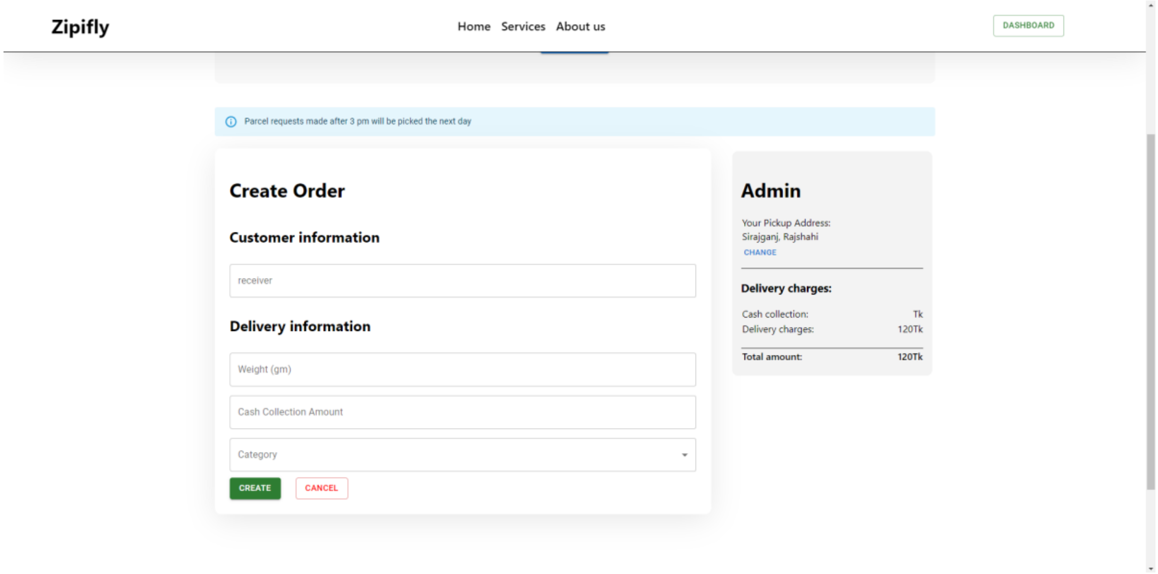
5.2.8 Create Parcel



Our rider will deliver your parcel without reveling your location

Figure 5.2.8: Create parcel

5.2.9 Create Parcel Form



Parcel requests made after 3 pm will be picked the next day

Create Order

Customer information

receiver

Delivery information

Weight (gm)

Cash Collection Amount

Category

Admin

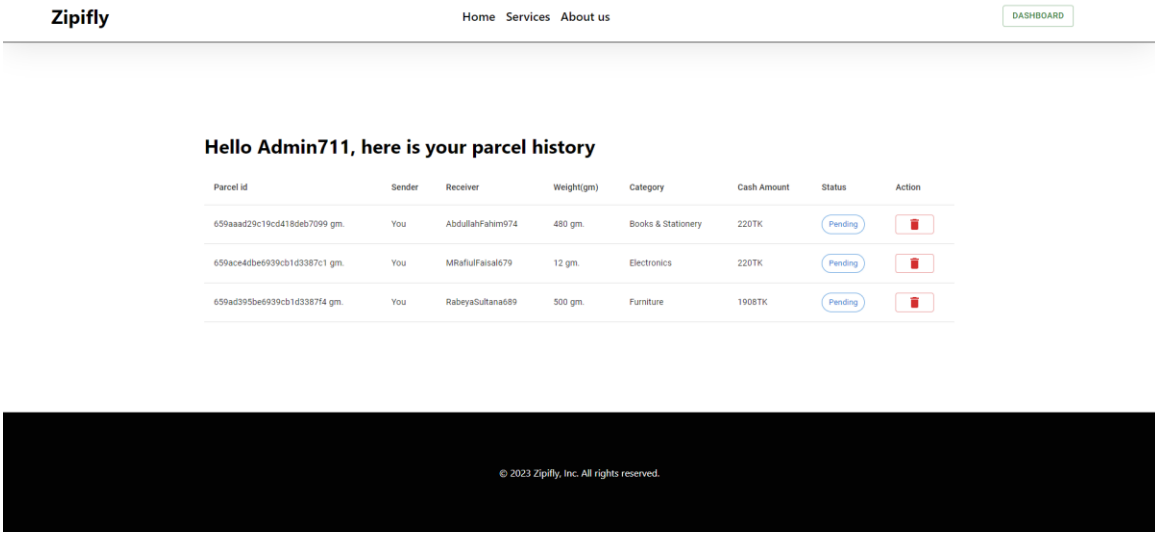
Your Pickup Address:
Sirajganj, Rajshahi

Delivery charges:

Cash collection:	Tk
Delivery charges:	120Tk
Total amount:	120Tk

Figure 5.2.9: Create parcel form

5.2.10 Display All Parcel



Hello Admin711, here is your parcel history

Parcel id	Sender	Receiver	Weight(gm)	Category	Cash Amount	Status	Action
659aaad29c19cd418deb7099 gm.	You	AbdullahFahim974	480 gm.	Books & Stationery	220TK	Pending	🗑
659ace4dbe6939cb1d3387c1 gm.	You	MRafiulFaisal679	12 gm.	Electronics	220TK	Pending	🗑
659ad395be6939cb1d3387f4 gm.	You	RabeyaSultana689	500 gm.	Furniture	1908TK	Pending	🗑

© 2023 Zipifly, Inc. All rights reserved.

Figure 5.2.10: Display all parcel

5.3 Testing Implementation

We meticulously assess the functionality, reliability, and security of the incognito parcel delivery app. Through a systematic testing approach, we validate the seamless integration of frontend and backend components, ensuring a robust, error-free, and secure user experience. The testing phase is instrumental in guaranteeing the app's adherence to design specifications, bringing us closer to a flawless and dependable final product.

5.4 Test Results and Reports

Table 5.4.1: Test cases and result

Test Case ID	Description	Test Steps	Result	Pass/Fail	Date
TC01	Registration	1.Navigate to Registration page	Successfully user registration	Pass	02.01.2024
		2. Enter valid input details.			
		3. Click Registration Button			
TC02	Login with valid credentials	1. Navigate to login page	Successfully user login	Pass	02.01.2024
		2. Enter Valid login details			
		3. Click login button			
TC03	Login with invalid credentials	1. Navigate to login page	Invalid username and password message given	Pass	02.01.2024
		2. Enter invalid login details			
		3. Click login button			
TC04	Track parcel by pid	1. Navigate to dashboard		Pass	02.01.2024

		2. Enter tid in the input field 3. Click track button	Parcel card component popup with parcel details		
TC05	Create Parcel	1. Click on create parcel button 2. Enter all information of parcel 3. Click create button	Parcel create successfully message shown	Pass	02.01.2024
TC06	Invoice with valid data from create parcel	1. Navigate to create parcel page 2. Enter all the information of parcel 3. Check if invoice is giving correct address and cash amount	Invoice generating correct address and cash amount	Pass	02.01.2024
TC07	All parcels of a user table	1. Navigate to all parcels page 2. See if the table is giving valid data	Table with valid data	Pass	02.01.2024
TC08	Delete parcel (if status: pending)	1. Navigate to all parcels page 2. See if the table is giving valid data with delete button 3. Click delete button	Delete successful message shown	Pass	02.01.2024
TC09	Delete parcel (if status is not pending)	1. Navigate to all parcels page	Can’t delete message pops up	Pass	02.01.2024

		2. See if the table is giving valid data with delete button			
		3. Click delete button			
TC10	Update parcel (if status: pending)	1. Navigate to all parcels page	Update successful message shown with updated data	Pass	02.01.2024
		2. See if the table is giving valid data with update button			
		3. Click update button			

CHAPTER 6

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

6.1 Impact on Society

The incognito parcel delivery app, Zipifly, stands to make a profound impact on society by addressing prevalent concerns surrounding privacy in the realm of parcel delivery. By safeguarding sensitive user information, it contributes to a heightened sense of security for individuals, particularly celebrities and influencers, who often face privacy challenges. The app's innovative features not only redefine industry standards but also foster a cultural shift towards prioritizing user privacy in the increasingly interconnected world of e-commerce. As Zipifly gains traction, it has the potential to set a new benchmark for responsible data handling in the delivery services sector, influencing a positive change in user expectations and industry practices.

6.2 Impact on Environment

While the primary focus of Zipifly is on user privacy and efficient parcel delivery, its indirect impact on the environment is noteworthy. By streamlining the delivery process and minimizing unnecessary data exposure, the app contributes to a reduction in paper waste generated from traditional delivery documentation. The optimization of routes and the efficient management of deliveries also hold the potential to decrease overall carbon emissions associated with transportation. As Zipifly gains widespread adoption, its environmentally conscious approach aligns with the global movement towards sustainable practices, making a positive contribution to reducing the ecological footprint associated with parcel delivery services.

6.3 Ethical Aspects

The ethical considerations surrounding Zipifly are integral to its design and implementation. The app prioritizes user privacy, aiming to eliminate the unethical practices of data misuse prevalent in the parcel delivery industry. By ensuring that sensitive information remains confidential, Zipifly addresses concerns related to unauthorized data sharing and potential misuse. Moreover, the app promotes transparency and user empowerment by allowing individuals, including celebrities and influencers, to control the exposure of their location and personal details. In navigating the intersection of technology and ethics, Zipifly sets a precedent for responsible and conscientious application development within the context of user privacy and data security.

6.4 Sustainability Plan

Zipifly's sustainability plan is rooted in a commitment to long-term viability and responsible growth. Key elements include:

Continuous Improvement

Regularly update the app to incorporate the latest security measures, technological advancements, and user feedback. Continuous improvement ensures the app remains resilient to emerging threats and aligns with evolving user needs.

Environmental Considerations

Explore eco-friendly options for packaging and delivery, minimizing the ecological impact of the service. Collaborate with environmentally conscious partners and implement green practices to contribute to sustainability.

Community Engagement

Foster a sense of community by engaging users in feedback loops, surveys, and forums. By actively involving users in the development process, Zipifly can adapt to changing preferences and ensure a user-centric approach.

Global Expansion with Local Sensitivity

Expand services globally while respecting and adhering to local privacy regulations and cultural sensitivities. Customizing the app's features and communication strategies based on regional nuances ensures a respectful and inclusive approach.

Ethical Data Practices

Uphold ethical data practices by reinforcing stringent security measures, promoting transparency, and adhering to privacy regulations. Maintaining user trust through responsible data handling is fundamental to the app's sustainability.

Social Responsibility Initiatives

Contribute to social causes and community welfare through targeted initiatives. By leveraging its platform, Zipifly can champion social responsibility, supporting local communities and addressing broader societal challenges.

Partnerships for Impact

Collaborate with like-minded businesses and organizations to amplify the positive impact. By forming partnerships that align with ethical and sustainability values, Zipifly can contribute to a network of responsible practices within the industry.

This sustainability plan underscores Zipifly's commitment to enduring success, responsible business practices, and positive contributions to the environment and society.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Discussion and Conclusion

Zipifly Incognito Parcel Delivery App represents a significant breakthrough in addressing privacy concerns within parcel delivery. Unlike existing apps, Zipifly prioritizes extreme privacy, allowing users to send and receive parcels without compromising personal information. The innovative design, secure onboarding, and user-friendly features distinguish Zipifly in the competitive market, catering specifically to privacy-conscious users. However, challenges include intricate algorithm development to balance extreme privacy and efficient delivery. Encouraging user adoption and building trust are critical, demanding targeted training and support.

In conclusion, Zipifly is a promising solution to pervasive privacy issues in the delivery sector. Successfully navigating algorithm complexities and security measures, it sets a new standard for user-friendly and private parcel delivery. Realizing the vision to redefine industry standards, Zipifly empowers users, reduces spam, and positions itself as a transformative force in the parcel delivery sector. This success marks the beginning of a new era where privacy and convenience harmonize to reshape the user experience in parcel delivery.

7.2 Scope for Further Developments

The successful implementation of the Zipifly Incognito Parcel Delivery App lays the foundation for a continuous journey of innovation and enhancement. The initial phase of development has addressed crucial aspects of privacy, efficiency, and user experience. However, to further elevate the app and respond dynamically to evolving user needs and technological advancements, several areas for future development are identified:

Enhanced Privacy Features

Explore and implement advanced privacy features, such as additional encryption protocols and secure communication channels, to provide an even higher level of confidentiality for user data.

Machine Learning Integration

Incorporate machine learning algorithms to analyze user preferences and delivery patterns, optimizing the incognito parcel delivery process and predicting user requirements for a more personalized experience.

Blockchain Integration

Explore the integration of blockchain technology to enhance the security and transparency of transaction records, providing users with an immutable and verifiable history of their parcel deliveries.

Smart Parcel Packaging

Integrate IoT (Internet of Things) devices for smart parcel packaging solutions, allowing users to monitor and control the condition of their parcels in real-time, including temperature, humidity, and security features.

Global Expansion

Extend the app's reach to a global audience by incorporating support for multiple languages, currencies, and international delivery standards, making Zipifly a versatile and accessible solution worldwide.

Collaborations with E-commerce Platforms

Form partnerships with major e-commerce platforms to seamlessly integrate Zipifly as a preferred and integrated parcel delivery solution, creating a symbiotic relationship that benefits both users and businesses.

User Feedback and Continuous Improvement

Establish a robust system for gathering user feedback and actively implement iterative updates based on user suggestions and evolving industry trends, ensuring the app remains at the forefront of innovation.

Accessibility Features

Enhance accessibility features, catering to users with diverse needs and ensuring a user-friendly experience for individuals with disabilities.

These future developments present exciting opportunities to further elevate Zipifly, making it a dynamic and cutting-edge solution that anticipates and adapts to the ever-changing landscape of parcel delivery and user expectations.

References

[1] Talukder, M. A. I. Mahmud, A. Sultana, T. H. Pranto, A. B. Haque, and R. M. Rahman, "A customer satisfaction centric food delivery system based on blockchain and smart contract," J. Inf. Telecommun., vol. 6, no. 4, pp. 501-524, 2022.

[2] H. Huang, A. V. Savkin, and C. Huang, "A new parcel delivery system with drones and a public train," J. Intell. Robotic Syst., vol. 100, pp. 1341-1354, 2020.

[3] Kreisel, D. K. (2003). "Incognito, Intervention, and Dismemberment in 'Adam Bede'." ELH, 70(2), 541-574.

[4] F. Zhou and J. Zhang, "Freight transport mode based on public transport: Taking parcel delivery by subway as an example," in Sixth International Conference on Transportation Engineering, Reston, VA, USA, Sep. 2019, pp. 745-754, American Society of Civil Engineers.

[5] ReactJS: A JavaScript library for frontend development, available at <<https://react.dev/>>, last accessed on 15 January 2024

[6] MongoDB and Mongoose: NoSQL database solution and mongoose is a ODM library for NodeJS, available at https://www.mongodb.com/docs/ , <<https://mongoosejs.com/>>, last accessed on 15 January 2024

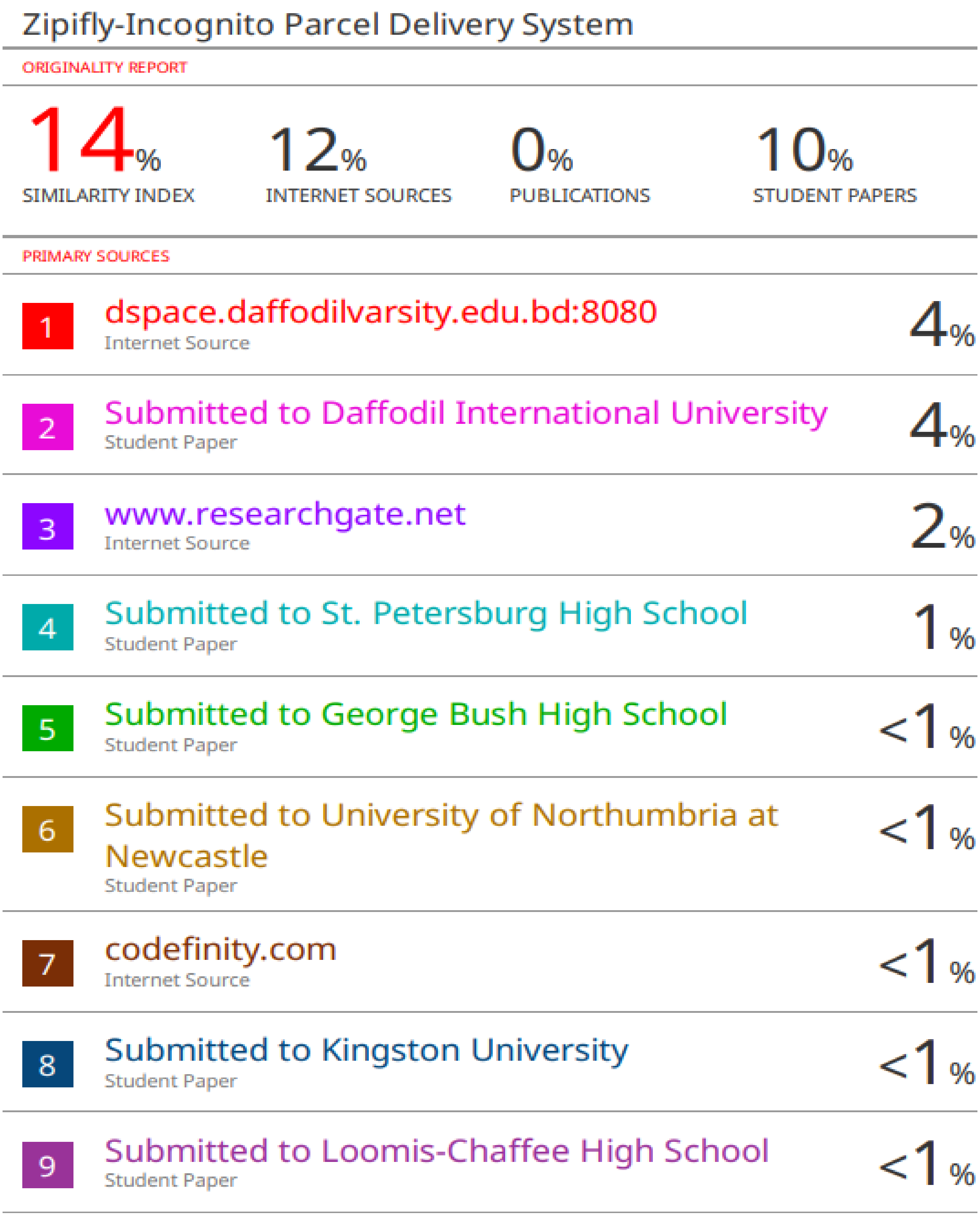
[7] NodeJS: It is a run time environment for JavaScript, available at <<https://nodejs.org/docs/latest/api/>>, last accessed on 15 January 2024

[8] ExpressJS: Framework for NodeJS and use for backend development, available at <<https://expressjs.com/en/5x/api.html>>, last accessed on 15 January 2024

[9] MaterialUI: A CSS framework for component development, available at <<https://mui.com/material-ui/getting-started/>>, last accessed on 15 January 2024

[10] Calemeo 2024: Global Parcel Delivery market report 2024, available at <<https://www.calameo.com/read/0074372095319e619a98f>>, , last accessed on 13 July 2023

[11] Parcel Delivery Market Report 2024 (Global Edition), available at <<https://www.cognitivemarketresearch.com/parcel-delivery-market-report>>, last accessed on 13 September 2023



Zipifly-Incognito Parcel Delivery System

ORIGINALITY REPORT

14%	12%	0%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	4%
2	Submitted to Daffodil International University Student Paper	4%
3	www.researchgate.net Internet Source	2%
4	Submitted to St. Petersburg High School Student Paper	1%
5	Submitted to George Bush High School Student Paper	<1%
6	Submitted to University of Northumbria at Newcastle Student Paper	<1%
7	codefinity.com Internet Source	<1%
8	Submitted to Kingston University Student Paper	<1%
9	Submitted to Loomis-Chaffee High School Student Paper	<1%