

Enhancing Quality Assurance in IT: A Case Report of Munchies

by

Saima Rahman Tasnia
20101274

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
February 2025

© 2025. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

A handwritten signature in black ink that reads "Saima". The signature is fluid and cursive, with the 'S' being particularly large and stylized.

Saima Rahman Tasnia
20101274

Approval

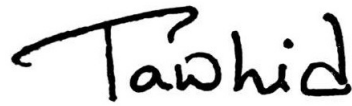
The thesis/project titled “Enhancing Quality Assurance in IT: A Case Report of Munchies” submitted by

1. Saima Rahman Tasnia (20101274)

Of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on February 03, 2025.

Examining Committee:

Supervisor:
(Member)



Md. Tawhid Anwar
Senior Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, Ph.D.
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

As a student of BRAC University, I was appointed as an intern at Munchies which is a top-tier food delivery service company that provides late-night delivery services of foods and necessary goods around Dhaka, Bangladesh. I was a remote employee and worked as a Quality Assurance Software intern in the IT sector to help enhance software quality by applying testing methods. My focus was to deploy mainly on manual testing systems while setting quality benchmarks and identifying current system problems to benefit both my studies and career.

Keywords: Late-night delivery services; Remote employee; Quality Assurance Software; Software quality; Testing methods- Manual testing; Current system problems

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Table of Contents	iv
List of Figures	vi
1 Introduction	1
1.1 Overview of Internship	1
1.2 Purpose of this Report	1
1.3 Goals and Objectives	1
1.3.1 Primary Aim	1
1.3.2 Specific Objectives	2
1.4 Research Methodology	2
1.4.1 Data Collection	2
1.4.2 Analysis Approach	2
1.4.3 Testing and Bug Identification Process	3
2 Company Overview	5
2.1 Company Profile	5
2.2 Mission and Vision	5
2.3 Operational Framework	6
2.4 Service Offerings	6
2.5 Recruitment Procedures	7
2.6 Ongoing Projects	7
3 Training and Induction	8
3.1 Orientation and Initial Training	8
3.2 Work Frameworks	8
3.3 Learning Resources	9
3.4 Team Dynamics and Communication	9
3.5 Organizational Culture	9
4 Remote work and data management	10
4.1 Work Overview	10
4.2 Bug Tracking	11
4.2.1 Full assignments(all-purpose)	11

4.2.2	Website	12
4.2.3	Customer App	13
4.2.4	Fields	13
4.3	Common Issues and Solutions	14
4.3.1	Rider App showed an incorrect balance amount at the start of 9 PM	14
4.3.2	Our Team Modified The Feature Update To Let Customers Spend More Money Through Each Order (Feature Update)	14
4.3.3	The Customer App on iOS suffered from a slide background activation problem	15
4.3.4	The Rider App showed users outdated information because it took time to refresh rider location data	16
4.3.5	Our system falsely identifies legitimate customers who cancel their orders as attempting fraud	17
5	Workflows	18
5.1	Order flow	18
5.1.1	General workflow	18
5.1.2	Customer-Admin-Rider Connection	19
5.2	Payment method	20
5.3	Rider Attendance	20
5.4	Workflow Optimization Strategies	21
5.5	Challenges in Order Flow and Solutions	21
6	Testing Experience	22
6.1	Engagement with Projects	22
6.2	Environment Setup and Tools	22
6.3	Testing Procedures and Protocols	22
6.4	Navigating Remote Work with Digital Tools	23
7	Conclusion and Reflections	24
7.1	Summary of Experiences	24
7.2	Lessons Learned	24
7.3	Future Aspirations	24
	Bibliography	26

List of Figures

2.1	Munchies-logo	5
2.2	Munchies- Web interface	6
2.3	Munchies- App interface	6
4.1	Bug Tracking:Full assignments(all-purpose) ₁	11
4.2	Bug Tracking:Full assignments(all-purpose) ₂	12
4.3	Bug Tracking:Website	12
4.4	Bug Tracking:Customer App	13
4.5	Bug Tracking:Fields	13
5.1	General workflow	18
5.2	Customer-Admin-Rider Connection	19
5.3	Payment method	20
5.4	Rider Attendance	20

Chapter 1

Introduction

1.1 Overview of Internship

Internships provide the most important bridge between the classroom and all — that is all — the life that students will one day have in the workforce. As a student of BRAC University, during my last few semesters, I decided to do my internship in Munchies instead of choosing a traditional thesis, to learn and work in the field of Quality Software Assurance (QA) since I always wanted to gain knowledge practically over theoretical experience. A popular food delivery service, which is Munchies Bangladesh, offers you late-night food delivery in Dhaka which would be a perfect fit to make you understand how exactly software development and QA process work. I was mainly appointed to work remotely; still, I was exposed to different parts of the software development life cycle, navigating potential pitfalls, and architecting solutions to guarantee the reliability and performance of our software. The practical project experience applied fundamental theory while establishing a clear comprehension of essential software quality assurance principles.

1.2 Purpose of this Report

This report provides an overview of my internship at Munchies including a business structure overview of Munchies IT department alongside my assigned responsibilities. This report aims at giving an elaborate overview of the intern program, focusing on the learning prerequisites, challenges encountered and the professional skills acquired during the period. The report aims to: To give a detailed account of Munchies and its practices, an article on training and induction of the new interns, talk about the actual QA methodologies and tools the IT across all departments commonly use, to look back on the practical experiences and lessons learned during the duration of the internship.

1.3 Goals and Objectives

1.3.1 Primary Aim

This report is prepared with the main purpose of evaluating the activities and the learning which took place during my internship at Munchies. This comprises an

evaluation of the firm's functions, the function of IT division, as well as ventures accomplished during the internship.

1.3.2 Specific Objectives

The specific objectives of this report include: Writing a company report about Munchies with the information about working conditions and atmosphere in the organization; List of services provided when interning at Munchies; Describing the occupation in Munchies focusing on the IT department; Including the information about the learning activities and the projects accomplishing during the internship.

1.4 Research Methodology

1.4.1 Data Collection

The internship delivered real-time Quality Assurance experience through direct interactions with Munchies processes. The techniques used included:

- **Manual Software Testing Observations:** We designed and documented many test scenarios to check how software operates across different usage and condition setups. Our group identified duplicate bugs by following established exploratory discovery methods during testing.
- **Bug Tracking Systems:** The team used JIRA to track problems and establish regulation procedures for monitoring issue team performance levels. Our team logged bug information with descriptions and impact assessment together with reproduction methods to help development teams resolve problems more efficiently.
- **Data and Metrics:** We tracked our test data inside Microsoft Excel to monitor how often bugs occur alongside their impact to the system alongside response time data and solution performance outcomes.
- **Collaboration with Stakeholders:** Attended team meetings to see how everyone planned and assessed fixes plus new functionality updates.

1.4.2 Analysis Approach

The research team used thematic analysis to understand the information they collected. Key technical elements included:

- **Performance Metrics:** We validated system components via performance measures such as load times and response speeds as well as inspections into types of failures. Researchers compiled structured Excel reports to analyze their results.
- **Defect Categorization and Root Cause Analysis:** Our team grouped bowlers based on bug types (functional, UI, security, and performance) to find out problem sources. We applied RCA(root cause analysis) techniques to discover what really made system failures happen.

- **Regression Testing Protocols:** Our team built complete test cases to confirm that existing problem fixes stayed stable through updated releases. Then our developers or senior teams used Postman API tests to automatically run test scenarios that validated API responses across multiple situations.
- **User-Centric Feedback Analysis:** The team analyzed user feedback flowing from the app to determine testing areas based on user needs and app usability problems.

1.4.3 Testing and Bug Identification Process

At Munchies I worked as an intern and tested all applications including Rider App and Munchies API alongside the Customer App and other programs. Our team tracked software bugs with JIRA and customized Excel sheets to properly document and solve all reports. My testing job involved running predefined tests and telling developers where problems existed.

The following steps were used to identify and resolve bugs:

- **Bug Reporting:** We recorded every problem in detail following standardized formats. This database carefully documented all parts affected, the descriptions of each, and the extent of impact.
- **Creating Test Cases:** Our tests reflected standard user behavior, as well as extreme usage scenarios to simulate the way customers interact with the system. For each reported error, our team confirmed and studied for accuracy through standardized tests.
- **Collaborative Fix Implementation:** Mainly, developers and seniors were the ones who did all the solving, whereas our intern's job was to give suitable suggestions. But during meeting days, we joined with our developers to understand, all faults clearly before fixing them and confirming the fixes through additional testing.

Example Issues Handled: Rider balance reset at 9 PM in Figure 4.1, Customer order amount limit increase in Figure 4.2 and so on.

Tools Utilized

Excel Tracker: The team mainly used an Excel system to record and control the bugs properly. The system offered one place to enter issues and tag them based on which app they impacted while showing their progress toward completion. The tracker system recorded important data elements like who received the tasks and the test and issue status labels. The tracker system captured problems such as “Rider balance zero on 9 PM” in Figure 4.1 and “Slide background click issue” in Figure 4.2 which helped teams recognize and address priority issues properly. This program helped us monitor progress by presenting clear data without manual sorting needed during our internship period.

JIRA: Teams worldwide use JIRA to manage projects and track issues while following Agile principles for their development approach. Even though I didn't work with

JIRA for automation tasks the team used it to organize project tasks control development speed and document errors throughout the whole company. JIRA allowed us to connect defect repairs to related development work while showing real-time updates and helping us meet our deadlines. Our team tracked sprint progress using burndown charts because these reports helped us meet sprint delivery targets.

Postman: Postman is a powerfully built API testing platform that helps users and developers test API connections plus validate results and create API request simulations. The team used Postman to check if the issue “Field pickup is missing” in Figure 4.1 in the API impacted how data moved through the Munchies API. Postman helped test fixes faster by saving our API requests for later validation. So, its key technical features include:

- **Request Building:** Through its platform Postman allows users to test multiple HTTP functionalities such as GET, POST, PUT, DELETE, PATCH, and alternative actions. We can set up API header info plus modify URL parameters with JSON or XML data structures plus pick your authentication type including OAuth token, API key or Bearer tokens.
- **Collection and Environment Management:** Postman helps users create collections of API tests to keep their work organized and to share it successfully with the team. Defined environments help your same requests run through various setup stages directly from the source data without manual value entry.
- **Automated Testing with Scripts:** With Postman we can create JavaScript test scripts before and during our tests. Users can create scripts that do both initial setup operations and response testing. Built-in assertions help validate response values from four basic tests: status codes, response times, headers, and body content.
- **Mock Servers:** Postman creates mock backend servers for developers to test their front-end response behavior before backend systems are available.
- **API Documentation and Monitoring:** Within Postman we can easily monitor API functionality through automation and the tool builds documentation straight from our stored collections[4].

Chapter 2

Company Overview

2.1 Company Profile

Dhaka is blessed with plenty of food delivery services; One of the most popular late-night food delivery services in Dhaka is Munchies. Keeping the requirements of the on-demand food delivery domain in the frame, Munchies has established a stronghold in the local market sending delicious food on time. The company operates primarily via its user-friendly app, which brings customers delivery from a range of restaurants and eateries.



Figure 2.1: Munchies-logo

Munchies is a late night food and essentials business model, tackling a huge gap in the market. It breaches a sound technological stack that allows smooth and rapid service rendering. This commitment to excellence and customer satisfaction has been an integral element of their growth and success at Munchies.

2.2 Mission and Vision

Therefore, Munchies aims at providing an exemplary food delivery service by regressing from the key values of satisfying customers, timely food delivery and availability of food orders. The overall goal of the company is to provide convenient availability of food through delivery particularly at night while making sure that the clients enjoy the delivery services.

Munchies is another platform to serve the people food in night at their doorstep and the main goal is to expand the business across Bangladesh. The primary goal of the company is to establish a high-quality service that customers trust, developing and adapting enhanced technologies to offer solutions to clients' needs[1].

2.3 Operational Framework

Munchies is an application that allows consumers to connect with restaurant and food outlets of all ranges. An operational framework comprises some key elements: a solid technological infrastructure that enables the functions of the application as well as a range of cuisine options available through a wide network of partner restaurants and food joints. A group of brilliant developers, QAs, and customer agents working together to provide you the best product and service possible. It has a mobile app that is user-friendly and simple to use, allowing customers to place purchase orders and track their order status with real-time mapping. So, a dedicated team does handles the backend processes for ensuring high-performing and seamless running of the app offering them a rich experience.

2.4 Service Offerings

Munchies is an all in one food delivery provider covering all your favourite cuisines and restaurants. It enables the clients to order food in a simple way and browse through tracking of deliveries on a real time basis Figure 2.2. Partnering with local Aside from the usual food delivery feature, Munchies also dabbles into delivery of other than food essentials, catering to the cravings of customer even during the wee hours of the night Figure 2.3. This service diversification has enabled the company to significantly increase its customer base and to achieve growth rates that are above the market average. With that being said, the company has a dedication to the excellence and happiness of its customers and that is, definitely, represented in its ability to offer services geared towards changing with the times that suiting the needs of all their clients[2].

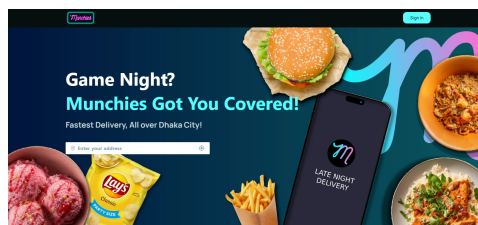


Figure 2.2: Munchies- Web interface

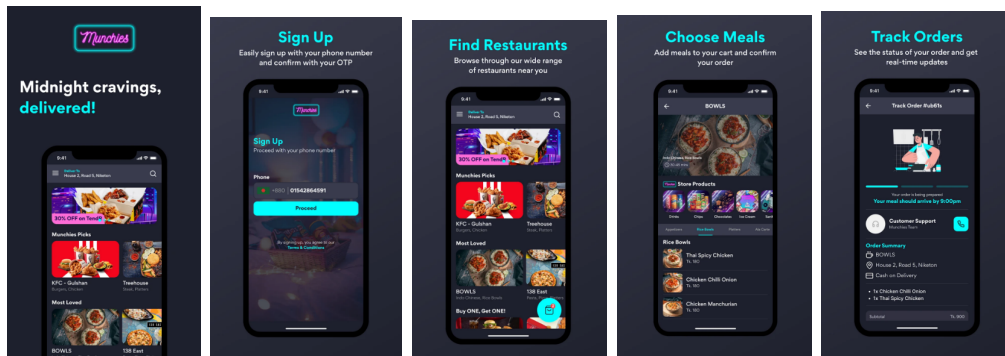


Figure 2.3: Munchies- App interface

2.5 Recruitment Procedures

A number of procedures exist in Munchies that are followed when hiring the employees in the company. Potential applicants are expected to provide a resume, a cover letter, then go for preliminary interview. The technical interview serves to evaluate three key areas during the process: candidates' problem-solving capacity alongside their company-relevant technological understanding and teamwork skills. Applicants move through an organized selection process to find professionals who meet both technical requirements and demonstrate proper organizational fit. Candidates move from a technical interview to face their potential hiring manager for an assessment of behavioral and cultural compatibility. Moreover, new recruits go through an orientation and training to increase their readiness for workstation assignments.

2.6 Ongoing Projects

Munchies has a number of active projects to innovate its service offering and grow its market share. Key projects include:

Mobile App Improvements: Ongoing improvements to the mobile application to optimize usability, introduce new functionality and fix bugs[3].

Delivery Optimization: The practice of building algorithms to improve the delivery process, draw out routes, reduce delivery time by making the process more operationally efficient.

Customer Loyalty Program: Designing a loyalty program to incentivize repeat purchase and enhance user retention.

Integration with Payment Gateways: Adding your payment gateways (both local and international).

Data Analytics and Reporting: Improving data analytics to better understand customer behavior, streamline operations, and make data-driven decisions.

Chapter 3

Training and Induction

3.1 Orientation and Initial Training

When I joined Munchies I was put through the groundside orientation which is designed to give new interns an inside look at the company its operations and how things are done and the culture and values of the company. Training in the beginning consisted of;

Introduction Of The Company: Slides on what munchies' vision and mission, and how they operate.

Departmental briefings: Organisation chart Reading through the roles and responsibilities of the department heads.

Critical Briefing and Working of IT Department IT Overview Pros: Structured Frame work of IT dept, Key Projects, anual Radar views and results carried out.

Software Tools and Systems: An overview of the software tools and systems (e. g. Postman API testing, JIRA, Trello and company specific software) used at the company.

3.2 Work Frameworks

The training provided by QA team of munchies was very detailed and it covered various methodologies and best practices which are very much important when it comes to software quality. It focused on manual testing practices in which testers had to discover and report defects in software that was being tested through well-planned, methodical testing approaches. One more major part was automated testing, where I learned the usage of tools that help us to test our code automatically such as Postman API testing, and Trello, JIRA (increasing the efficiency and accuracy of the tests done automatically where we can simulate everything the user does and repeat a specific task n number of times).

A large part of the training was the development of test cases, where one created exhaustive schedules that covered all possible parts of the making of software; This step insured that everything that a user could do to the software was tested.

Bug Tracking and Reporting: Moreover, bug tracking and getting reported were a mandatory part and for those, they use Trello, and JIRA, it is a sturdy issue tracking tool that allows you to log and keep track of bugs at short intervals in the development lifecycle. At every step, a detailed documentation of bugs was fed

back and regular updates were taken to make sure those issues were fixed within the defined timeframe.

3.3 Learning Resources

Munchies had a lot of resources in place to help me learn and grow. These were in enrollments in remote courses obviously in QA and obviously in development, these were other online courses (ie Coursera and Udemy) Also, I found detailed documentation, manuals, literally everything to get better at my work, and also for me to learn all the technical part that was needed to do my job. We also scheduled mentorship sessions with our senior QA engineers, and I could to share stories, ask for advice, and have a glimpse on how things are done in the industry.

3.4 Team Dynamics and Communication

In IT, communication and teamwork are key. I actively engaged in a range of primary experiences enabling collaboration and knowledge transfer during my internship. Weekly stand-ups were brief meetings conducted each day to review what was done and what was next. Sprint planning was a participatory meeting to plan tasks and set priorities for each sprint cycle. Retrospective: A meeting at the end of each sprint to review performance, identify any improvements, and celebrate the contributions of the team. I also collaborated cross-functionally with developers, product managers, and other stakeholders and ensured appropriate alignment of others' work and that risks were made apparent and addressed quickly.

3.5 Organizational Culture

Munchies create a friendly environment of change, networking, and constant evolution. The organizational culture is set on the following: encouragement of open and transparent communication on all levels of the organization. Regular team meetings, team-building events, cross-functional projects contribute to building a collaborative and team-working culture. Munchies also continually reminds their team to contribute with ideas for improving methods and products, promoting the inquisitive nature of the people who work at the startup.

Chapter 4

Remote work and data management

4.1 Work Overview

Remote work is an important part of the interns' responsibilities at Munchies because it helps us work on various tasks from different locations. It is still flexible enough to pursue other duties that we have as we continue to be productive. Communication is a crucial element of our process, and data management is perhaps even more valuable since it helps oversee all projects and recognize possible problems. We use many things, such as Microsoft Excel, in order to organize tasks, monitor data, and bugs' reports. The particular tools enable the documenting of problems that are likely to emerge as the software is being tested besides aiding the proper organization of information such that it can be easily retrieved by all members within the team. This kind of work arrangement requires no gaps in the process of sending and receiving information and experiences; therefore, we keep things going with updates and document exchange. Such a structure enables us to efficiently address any of the emerging issues, despite our locations, thus maintaining a healthy performance.

4.2 Bug Tracking

4.2.1 Full assignments(all-purpose)

	A	B	C	D	E	F
1						
2		Bug Tracking		Munchies bug tracking for all purpose		
3		Issue on	Assignment	Assignie	Test Case	Status
6		Rider App	Rider balance zero on 9 PM	Saima	Passed	Done
7		Rider App	Date range data validation in history	Hasan		Not started
8		Rider App	Rider app invoice generation	Saima		Not started
9		Munchies API	Field pickup is missing	Saima		Not started
10		Munchies V2	The order page is too loading	Saima		Skipped
11		Munchies V2	On the wallet page, today's collection couldn't update. Collection amount remains 0.	Hasan	Failed	Done
12		Rider App	Unable to login			
13		Rider App	After logout,why I get the notification for mirza rider	Saharukh	Passed	Done
14		Rider App	Delivery Done without Mark Pickup and Mark Delivery.			
15		Rider App	After pickup add a call button.		Passed	
16		Rider App	After completing a delivery rider didn't show km in the Km graf	Faisal		
17		Rider App	The pick order didn't count, the ongoing order field didn't update.			
18		Rider App	Order assigned but didn't show the order in app loading the page almost 10 minuties	Saima		
19			History should be updated. Delivery cod or Bikash, these are not seen.	Saharukh	Passed	Done
20			Restaurant location issue.	Saima		Skipped
21			Rider want a whatsapp call button	Saharukh	Passed	Done

Figure 4.1: Bug Tracking:Full assignments(all-purpose)₁

These includes all the bugs identified on the site, mobile app and all internal intelligent software tools we use in the organization in Figure 4.1 and 4.2. Each bug identified is ranked depending on the extent of the problem it causes to the experience of the end-user who is aimed to be addressed by the development team first.

	A	B	C	D	E
1	Bug Tracking				
2	Issue on	Assignment	Assignie	Test Case	Status
16	Madmin	↓ In Aarchive order added some search filter	Jawad	↓ Passed	↓ Done
17	Feature	↓ Customer mark as fraud if the user is not unreachable or cancel order. Also added a section to see record of mark customer	Saima	↓ Passed	↓ Done
18	Update	↓ Customer order amount limit is increasing 5000 to 8000	Saima	↓ Passed	↓ Done
19	Customer App los	↓ slide backgroud click issue. E.x = when i click around back button munchies store is clicking and show the menu list			
20	Partner App	↓ Rider app location issue in map rechecked	Saima	↓ Passed	↓ Done
21		↓ Promotinal banner title gradient issue	Mirza	↓ Passed	↓ Done
22	website	↓ Instagram link issue in website footer	Jawad	↓ Passed	↓ Done
23	Madmin	↓ User Deactivation Block	Faisal	↓ Passed	↓ Done
24	Feature	↓ Analytics Orders->Dashboard	Saima	↓ Passed	↓ Done
25	Customer App	↓ Remove Reffer validation	Faisal	↓ Passed	↓ Done
26	Partner web	↓ Rington Change	Jawad	↓ Passed	↓ Done
27	website	↓ User location will be displayed in alphabetical order	Saima	↓ Passed	↓ Done
28	Partner App	↓ Rington Change	Saharukh	↓ Passed	↓ Done
29	website	↓ Product View more Image change into size 1:1	Mirza	↓ Passed	↓ Done
30	Partner	↓ Resturant Availability button added	Hasan	↓ Passed	↓ Done
31	website	↓ Product disscunt issue	Saima	↓ Passed	↓ Done
32	website	↓ Rating -1 issue sloved	Saima	↓ Passed	↓ Done

Figure 4.2: Bug Tracking:Full assignments(all-purpose)₂

4.2.2 Website

	A	B	C	D	E
1					
2	Bug Tracking				
3	Issue on	Assignment	Assignie	Test Case	Status
7	website	↓ After changing the order status estimated time is not change	Saima	↓ Passed	↓ Done
8	Promo	↓ Welcome Promo for free delivery for new user ↓ Test scenario: The product price is less the delivery charge.		↓ Passed	↓ Done
9					
10					
11					
12		↓ Section			
13		↓ Duplicate Profile Value			

Figure 4.3: Bug Tracking:Website

In Figure 4.3, defects such as usability, navigation, and functionality of a website are key areas of bugs in a website. In general, the focus is on what end-users need as they have to be able to browse for products, make an order, and track delivery.

4.2.3 Customer App

	A	B	C	D	E
1					
2	Bug Tracking				
3	Issue on	Assignment	Assignie	Test Case	Status
6	Customer App	After changing the order status estimated time is not change	Saima	Passed	Done
7	Customer App	If the restaurant is off, the search will show Restaurant Is Not Available	Saima		

Figure 4.4: Bug Tracking:Customer App

In Figure 4.4, the customer app bugs typically include mobile-only problems, including compatibility with distinct operating systems, frequent app crashes, and slow loading speed. Munchies always enhances the application to address such issues and make difficult interfaces easier to navigate.

4.2.4 Fields

1	
2	App
3	Which app the issue is still on
4	
5	Madmin
6	website
7	Server
8	Customer App
9	Partner App
10	Rider App
11	Sunmi Device
12	Munchies API
13	Munchies V2
14	Rider MS Websocket
15	Feature

Figure 4.5: Bug Tracking:Fields

With reference the use of bug tracking, the term “Fields” In Figure 4.5 is the space where entry is made in Munchies including the order form, customer’s details, his/her address, and payment details. These fields are very important in making sure that all data that is input in the site both by the users and the site administrators is well captured. Problems in such fields may range from improper validation, incorrect data handling, formatting errors and may hinder the flow of information and this may lead to wrong order information, late delivery or inability to process payments among others.

To keep the experience smooth, Munchies also pays particular attention to bug reports connected to these fields, always resolving them, so the fields work correctly on all platforms. It is also best-practice protective approach that prevents more mistakes in data input, enhances interconnected communication between customers, administrators, and delivery riders, and increases overall platform stability in functionality and customer value.

4.3 Common Issues and Solutions

4.3.1 Rider App showed an incorrect balance amount at the start of 9 PM

- **Problem:** The rider's balance stayed at zero at 9 PM when they had balance history.
- **How the System Worked:** The rider's balance reflected changes from completed deliveries as well as the payouts received. The backend engine performed calculations on income and costs together with adjustments which modify the final balance value. The system operated scheduled maintenance during 9 PM through the execution of scheduled processes for balance refresh operations.
- **How the Bug Was Identified:** During manual testing activities testers or riders confirmed that the system reset balance information precisely at 9 PM. Database and API Logs revealed wrong balance change records during this period. After the balance updates testers conducted regression testing which revealed wrong previous balance information. Riders noticed their income showed incorrect amounts which triggered further analysis on the matter.
- **Possible Causes:** A back-end system omits correct earnings computations since it contains a time-based job malfunction. API failed to retrieve the right balance information stored in the database. The front-end produces incorrect balance notifications because of an active cache system.
- **Test Cases Applied:**
 - tracked how balance updates changed between scheduled times of day.
 - tested our balance update system both at 9 PM and just ahead and behind that time.
- **Solution:** The team fixed the time-based problem that made the system reset balances too early and updated the balance retrieval system to pull accurate database information.
- **Outcome:** The team fixed the problem which made balance readings more precise and helped riders better understand their account status.

4.3.2 Our Team Modified The Feature Update To Let Customers Spend More Money Through Each Order (Feature Update)

- **Problem:** The order amount restriction should go from 5000 to 8000 as part of our changes.
- **How the System Worked:** As a customer control feature, the system established maximum order values. The system started accepting greater order amounts after the update implementation. The modification probably influenced three essential elements of checkout processing as well as payment gateway operations and API limitations.

- **How the Bug Was Identified:** Testers executed test cases through making orders which exceeded previous order limits in order to verify system behavior under these circumstances. Back-end and API Validation confirmed that the system displayed the updated spending restrictions correctly in database and API communication. Customers who reported issues placing orders above the previous limit could have filed reports to the team.
- **Possible Causes:** Payment gateway has the potential to decline transactions that surpass previous funding limitations. Users may encounter outdated limit displays on their screens which leads to misunderstanding. The new spending rules might generate inconsistent results because API fails to provide accurate validation.
- **Test Cases Applied:**
 - testing confirmed that the system accepted orders that did not exceed 8000.
 - tested placing orders greater than 8000 items to identify validation failures.
 - system transition needed to work without causing problems between 5000 and 8000 orders.
- **Solution:** Our team updated how the system validates order amounts to support both the new 8000 limit and preserve system speed.
- **Outcome:** The feature update went live as expected to provide customers with more options in their orders.

4.3.3 The Customer App on iOS suffered from a slide background activation problem

- **Problem:** When tapping close to the back button users would interact with background features and select the wrong menu options.
- **How the System Worked:** When users navigated to different apps on iOS the current app moves toward background status. Different types of applications required either background process pausing or real-time updates to continue working. The bug triggered accidental activation that result in app reactivation during periods where activation should not occur.
- **How the Bug Was Identified:** IOS users noted unusual application actions through their complaints. The testing confirmed the problem existed only within the iOS system while Android remained unaffected. The developers used event logs and debugging methods to inspect iOS lifecycle events regarding application switch actions. The system logged indicated potential unexpected API background activity during monitoring.
- **Possible Causes:** A software error exists in iOS's application lifecycle management system. The application fails to suspend its network communication when users switch to other apps. The product's mechanism intended for automated updating inadvertently set off background functionality.

- **Test Cases Applied:**

- tested which part of the back button could accurately receive taps.
- tested button and background taps to find overlapping screen actions.

- **Solution:** We adjusted the touchable zone on the back button to make it easier to handle and prevent accidental actions.

- **Outcome:** Better performance and smoother navigation with fewer operation problems for users.

4.3.4 The Rider App showed users outdated information because it took time to refresh rider location data

- **Problem:** The map tracked old rider positions because the GPS information arrived late.

- **How the System Worked:** Throughout regular intervals the Rider App transmitted location data to its server. The backend system processed received updates before sharing them to customers and admins. During this bug the rider location displayed outdated information because the refresh rate was slower than expected.

- **How the Bug Was Identified:** The test team conducted direct observation tests to monitor GPS tracking in real time alongside display positioning information. Checked rider location update frequency in API logs. The debugging procedure checked how effectively the device sent location requests to keep the communication channels active with the server. Riders/customers notified us through complaints about tracking delays.

- **Possible Causes:** An excessive length of time (five minutes instead of ten seconds) specified for location update intervals caused the problem. The API experienced delays in processing instant live updates. The application showed outdated locations because of caching problems while avoiding new location retrievals.

- **Test Cases Applied:**

- tested how quickly GPS data updated in real-time.
- checked if the product display area match real locations and shows updated maps properly.

- **Solution:** Our system updates rider positions more often through optimized API calls to show their most recent tracked locations.

- **Outcome:** The system showed delivery updates precisely as they happened to provide correct tracking status.

4.3.5 Our system falsely identifies legitimate customers who cancel their orders as attempting fraud

- **Problem:** The system marked valid payments fraudulent when customers became unreachable or canceled their orders.
- **How the System Worked:** The detection system tracked patterns of cancellations through its monitoring functionality. The system marked potential fraud when customers perform multiple cancellations. The system bug identified genuine cancellation requests as fraudulent activities.
- **How the Bug Was Identified:** Users reported experiencing problems with system blocking them while trying to order. The developers examined the factors which caused the system to identify standard users as potential fraud cases through algorithm analysis. A backend evaluation of database structures discovered that technical cancellation conditions generated numerous incorrect fraud alerts. The execution of test cases included creation of various scenarios which reproduced the identified problem.
- **Possible Causes:** The existing fraud detection threshold operated at a point which led to incorrect fraud identifications when customers canceled their orders twice (e.g. canceling 2 times flagged as fraud). Without proper distinction the system failed to distinguish between authentic and fraudulent cancellation events. The backend logic applied flawed flagging rules to its operations.
- **Test Cases Applied:**
 - tried test situations with customers who canceled orders on their own and system failure conditions.
 - system checks when customers follow normal procedures versus when they perform suspicious actions.
- **Solution:** The system design split fraudulent activity detection from order cancellation by creating special rules to spot unusual customer actions.
- **Outcome:** Enhanced customer satisfaction results from better ways to spot fraud.

Chapter 5

Workflows

5.1 Order flow

5.1.1 General workflow

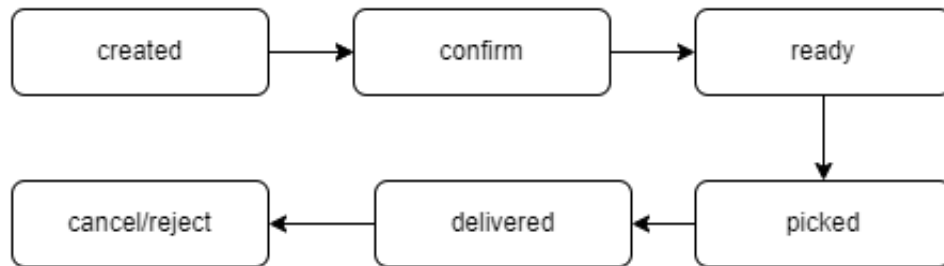


Figure 5.1: General workflow

At Munchies the order flow is integrated with system solutions to enable a standard and seamless procedure guaranteeing timely food delivery for all customers. The tale starts when the Munchies customer makes an order over their app or website. Upon receipt of the order, it is confirmed further by its admin team. Admin checks the availability of food items and assigns order to the nearest available rider Figure 5.1.

5.1.2 Customer-Admin-Rider Connection

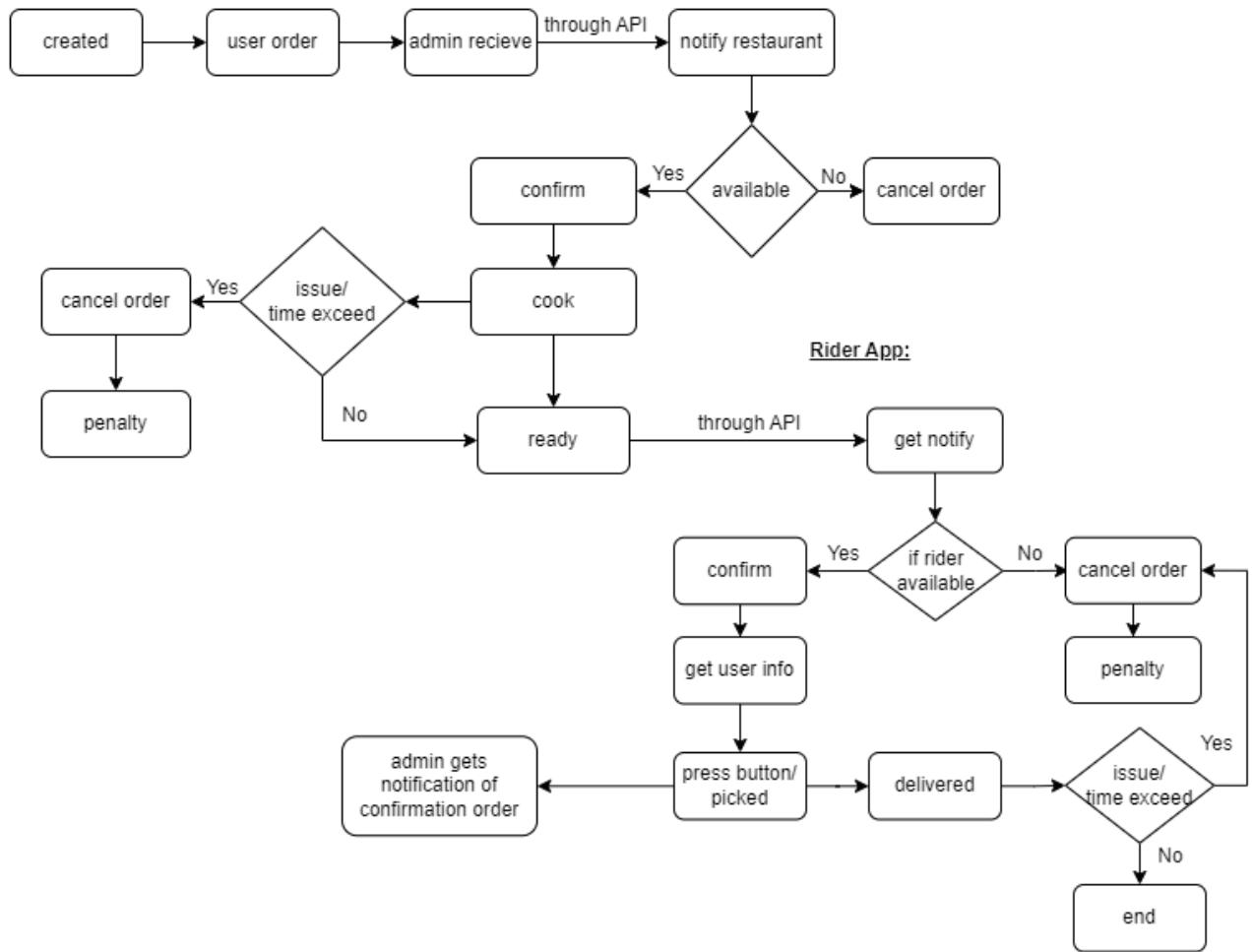


Figure 5.2: Customer-Admin-Rider Connection

This in Figure 5.2 workflow revolves around the customer administration-rider link. Communication and updates between all of these three entities are made in real-time to enhance order flow in the process. After assignment of the order, the rider takes the food from the restaurant and then delivers it to the door of the consumer. This work flow is backed by a solid back end architecture that keeps track of the order status and relays status to the customer which goes a long way in improving the customer experience.

5.2 Payment method

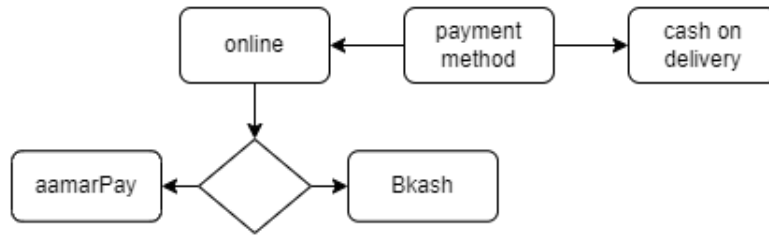


Figure 5.3: Payment method

Figure 5.3, Munchies accepts several types of payments to ensure that everyone has their favorite modes of paying for food. In terms of payments, customers have to options of going for electronic payment through credit or debit or through mobile money or by cash on delivery. The payment procedure is quite safe and SWOT incorporates efficient payment options that provide services for effecting payment smoothly without any hindrances. The system is capable of processing high volume of transactions particularly during a period of high patronage, while at the same time maintaining high level of security.

5.3 Rider Attendance

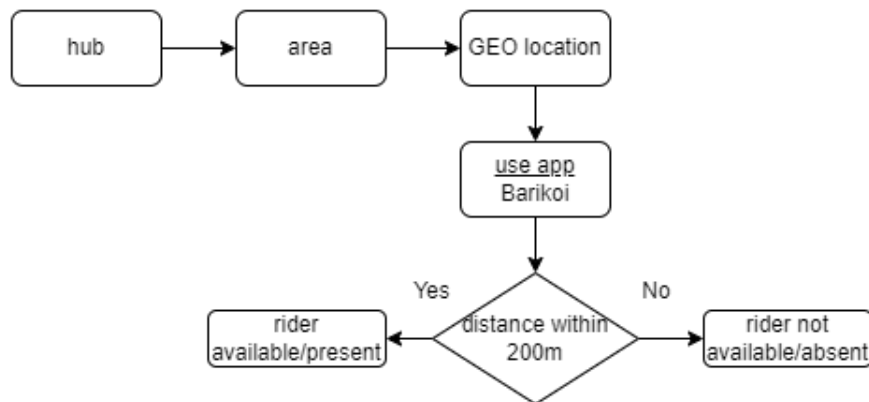


Figure 5.4: Rider Attendance

Figure 5.4, Perhaps the most important factor regarding the attendance tracking for Munchies' riders is the operational need. To ensure adequate availability of riders, the company uses an automatic check-in system to track actual attendance. The shift management of all the rider, which is done through this system, enables the admin team to ensure that all riders in the company are always available on shift during the delivery services. It also assist in evaluating the riders' performance and their working schedules with a view to improving their services delivery.

5.4 Workflow Optimization Strategies

Ordering is a big part of the work, and so, Munchies capitalize on ensuring that this section does not lack structure since that is the most vulnerable area to hitches. They manage to ensure adequate numbers of riders who are awake during busy periods that may be late at night, or during weekends. They also maintain a basic system of automating the assignment of each order to the closest rider to avoid slowing this process. Alternatively, through a proper planning and execution of these tasks, coupled with simple instruments, Munchies has made sure that their food delivery service is effective for any client.

5.5 Challenges in Order Flow and Solutions

Although, Munchies has designed efficient processes, it experiences several issues about efficient order flow. Foremost concerns are delayed supplies during festive seasons, incorrect order tracking, and payment gateway problems. In order to address these problems, Munchies has put into place a strong feedback mechanism with the possibility to report problems through the application. They have also created reorder strategies like emergency delivery routes and substitute payment methods in case of interferences. Moreover, since riders must maintain their orders' fluidity, effective training sessions for riders regarding suggested routes and dealing with customers are also necessary.

Chapter 6

Testing Experience

6.1 Engagement with Projects

While I was an intern, the interactions with projects were limited because I was assigned to work primarily from home and had to carry out data management tasks and software testing mostly. Some of the tasks that I performed on the day-to-day basis as an intern were as follows, managing and archiving bugs and reports, writing comprehensive test cases and evaluating software performance problems by using Microsoft Excel. In this context, first, I ensured the detailed documentation and structured data to help the development team in finding and solving software problems with effectiveness as a remote worker. It helped me provide input into the overall quality assurance process that a physical presence in the project environment would have facilitated.

6.2 Environment Setup and Tools

The testing location in this case was set remotely, and those involve the use of various tools in order to enhance information sharing. Postman API and JIRA are efficient tools to use for testing automation and bug reporting but the primary tool for handling and sorting all the work-related data in my case was Microsoft Excel.

For the remote working setup, Excel was used effectively in managing the test cases logged and recorded bugs among others. It enabled me to sort data, filter and analyze, thus facilitate the task of sharing information with a team to keep abreast with the task progress. As a way of working from home, this allowed for proper management of the tasks affecting the performance and communication with other individuals through constant update and passing on of the Excel sheets.

6.3 Testing Procedures and Protocols

Since my responsibilities involved mainly remote work, I followed testing procedures that prioritised documentations and reporting aspects. By employing Microsoft Excel, I faithfully followed different small-scale test cases and bugs' reporting and their resolution process. My responsibility included the reviewing, sorting and proper classification or documentation of all tests results in a way that would afford easy

identification and handling by the development team as problems arose.

This approach enabled me to be otherwise actively involved in quality assurance even though I rarely reported to the office. It also made it easier to consult with the members of the team to keep everyone up to date on the software's capabilities and the work being done on bug solving. By following these strict testing protocols and using a clean format of presentation suitable for excel it was easy to support the general goal of getting reliable software.

6.4 Navigating Remote Work with Digital Tools

Transitioning to a remote work required developing one's experience in the use of specific technologies that help fulfill certain tasks and projects. Some of the tools we used are bug trackers like JIRA and API testers like Postman. However, I concentrated on manual testing instead of the automated tools, however, tested solutions positively affected the communication and tracking of the work among the team members. Since using JIRA, it became possible to work with bugs, track problems, and keep all workers in the team aware of the ongoing situation. Likewise, there are Postman useful to check the presence of APIs to ensure the software functionality and identifying any differences. It helped also to maintain effectiveness of the quality assurance processes with which the collaboration was provided even in remote manner.

Chapter 7

Conclusion and Reflections

7.1 Summary of Experiences

I have my existing role going at Munchies being a good Quality Software Assurance as QA intern, where I learned something new every day within Software Testing and Quality Assurance. I have had first-hand experience with several testing tools and methodologies like automation testing(Postman API), bug tracking(JIRA) throughout my tenure here. I contributed in everything in software development life cycle like planning, test case designing, test case execution, defect management etc. I have helped in various projects like mobile app testing and performance evaluations through which I have improved my skills and understanding about key areas of software quality assurance. Also, this has provided me with the understanding and competence to perform QA principles in a practical context and therefore, guaranteeing quality work on software.

7.2 Lessons Learned

Working at Munchies has taught me some important lessons in a short period of time. The one is automation as we value that. Thus proved automated testing is required for efficient and accurate QA. Performing Automated Testing: Automation enables us to run more number of test cases using automation tools with more reliability in the application. Collaboration and communication is the other key take-away. A good team work is in the very base of successful project execution caused by a clear communication. Regular meetings like daily stand-ups and sprint retrospective have helped in smooth collaboration and quicker resolution of issues.

Moreover, the internship helped me to understand the idea of personal development as a vital aspect. It has let me continue to enhance my problem-solving skills — debugging involves problem-solving and adequate observation skills that accompany the analysis of minute defects in the software.

7.3 Future Aspirations

I feel that I would like to be a Quality Software Assurance tester after my internship at Munchies. From here on, I plan to enhance my knowledge through learning and getting awesome in my QA methodology and tools. This includes pursuing more and

more courses, attending workshops and staying on top of industry trends to supplement my skillset. I also have the ambition to help innovation with my knowledge by contributing in projects that would raise the bar in software quality or the end-user experience. The program also includes creating testing strategies, application of automation practices, and discovering new technology trends. Furthermore, Due to I deeply attach professionally right now and aspiring of taking myself to next level of software quality assurance. This may mean getting certifications, becoming part of professional groups, and meeting people involved in the industry. Through my internships at Munchies, these goals now have a solid foundation and I am thrilled to only expand and deepen them with my future career.

Bibliography

- [1] A. Mozumder, “Analysis of sustainability and competition of munchies in bangladesh,” 2022.
- [2] Munchies, *Munchies - late night food delivery in bangladesh*, accessed: June, 2024. [Online]. Available: <https://www.munchies.com.bd/>.
- [3] MunchiesBD, *Munchies - linkedin profile*, accessed: June, 2024. [Online]. Available: <https://bd.linkedin.com/company/munchiesbd>.
- [4] Postman Learning Center, *Postman documentation - overview*, accessed: January, 2025. [Online]. Available: <https://learning.postman.com/docs/introduction/overview/>.