UNIVERSITY OF FRONTIER TECNOLOGY, BANGLADESH (UFTB)

LAB REPORT

SUBMITTED BY:
MIZAN KHAN
RUKSANA AKTER ROJONY
S M REDOAN ULLAH RAHMAN

1ST YEAR 2ND SEMESTER

SUBMITTED TO:
SHIFAT ARA RAFIQ
LECTURER (UFTB)

DEPARTMENT OF
SOFTWARE ENGINEERING

FACULTY OF
SOFTWARE & MACHINE INTELLIGENCE ENGINEERING

LAB REPORT NO: 02

COURSE TITLE: System Analysis & Design

COURSE CODE: SE 118

LAB EXPERIMENT NAME: Software Requirements Specification Document.

LAB DATE : 19/08/2025
SUBMISSION DATE : 25/08/2025

LAB REPORT STATUS

Comments: ..... Signature: .....
Date : ..... Marks : .....

HomelyBites

SoloByte

Department of Software Engineering
UNIVERSITY OF FRONTIER TECHNOLOGY, BANGLADESH

September 2, 2025

Contents

1 Objective

The primary goal of ***HomelyBites*** is to build a digital platform that empowers housewives to become home-based entrepreneurs by selling homemade meals. It aims to promote healthier eating habits, strengthen local communities, and offer customers a reliable alternative to commercial food services. Through intuitive technology, secure transactions, and efficient delivery, ***HomelyBites*** bridges the gap between home chefs and food lovers, creating a socially impactful and scalable ecosystem.

2 Problem Analysis

Modern lifestyles have increased the dependence on restaurants and fast food delivery, often at the cost of health, hygiene, and affordability. People struggle to find home-style meals that are both nutritious and reasonably priced. Meanwhile, many housewives with strong cooking skills remain confined to their households, lacking opportunities to showcase their talents or generate income despite their abilities.

This creates a two-sided gap:

- Customers are deprived of authentic home-made food options.
- Housewives lack accessible platforms to monetize their cooking skills and contribute financially to their households.

HomelyBites addresses this gap by empowering housewives to become home chefs and providing customers with affordable, healthy, home-style meals. This not only ensures convenient access to nutritious food, but also promotes women's economic empowerment and community well-being.

3 Methodology

The development of ***HomelyBites*** follows an Agile Software Development approach, with iterative cycles to ensure continuous feedback, rapid delivery, and adaptability to user needs. The methodology integrates principles of Extreme Programming (XP) to enhance code quality and responsiveness, particularly through practices like Test-Driven Development (TDD), pair programming, and frequent releases.

Key Phases:

3.1 Requirement Analysis

- Stakeholder interviews and market research to define user needs.
- Prioritization of features based on business impact and feasibility.

3.2 System Design

- System Design Modular architecture planning for scalability and maintainability.
- UI/UX prototyping with accessibility and multilingual support.

- Agile sprints with daily stand-ups and sprint reviews.
- Implementation.

3.3 Testing and Quality Assurance

- Unit, integration, and regression testing.
- Security testing to ensure data protection and compliance.

3.4 Deployment and Maintenance

- Cloud-based deployment with CI/CD pipelines.
- Ongoing monitoring, bug fixes, and feature enhancements.

This methodology ensures that ***HomelyBites*** evolves with user feedback, maintains high software quality, and aligns with its social and business objectives.

4 Introduction

This Software Requirements Specification (SRS) defines the complete functional and non-functional requirements for ***HomelyBites***, a food delivery platform that connects home chefs with customers. It outlines the system's scope, which includes customer apps, chef dashboards, delivery coordination, and admin oversight. The document is intended for developers, project managers, testers, marketing staff, and documentation writers. It is organized into sections covering the product overview, system features, interface requirements, performance expectations, budget, and conclusion.

5 Project Scope

HomelyBites is a web and mobile-based food delivery system designed to connect home chefs—primarily housewives—with customers seeking fresh, healthy, and homemade meals. The platform enables chefs to manage menus, accept orders, and earn income from their kitchens, while customers benefit from nutritious alternatives to commercial fast food. ***HomelyBites*** supports corporate goals such as promoting social entrepreneurship, expanding market reach through hyperlocal engagement, and enhancing customer satisfaction by offering personalized, culturally diverse meal options. This SRS outlines the full scope of the ***HomelyBites*** system, including user interfaces, backend services, delivery coordination, and administrative oversight. If part of a broader ecosystem, this document focuses specifically on the core food ordering and delivery module and its integration with payment, tracking, and user management components.

6 Overall Description

6.1 Product Perspective

HomelyBites is a standalone web and mobile application. It differs from traditional food delivery platforms by focusing on home-cooked meals and empowering housewives as micro-entrepreneurs.

6.2 Product Features

- Home Chef Registration and Menu Management.
- Order Placement and Real-Time Tracking.
- Secure Payment Integration.
- Delivery Personnel Coordination.
- Mobile App Interface for Customers and Chefs.
- Admin Dashboard for Oversight and Analytics.

6.3 User Classes and Characteristics

6.3.1 Operating Environment

- Frontend: React.js (Web), Flutter (Mobile).
- Backend: Node.js/Express.
- Database: MySQL or MongoDB.
- Hosting: AWS or Google Cloud.
- Mobile OS: Android 12+, iOS 15+.
- Payment Gateway: Stripe or SSLCOMMERZ.

6.3.2 Design and Implementation Constraints

- Works well even with slow internet.
- Follows food safety and privacy laws.
- Can grow and handle more users in the future.
- Supports both Bangla and English languages.

7 System Features

7.1 User Login

- Secure login with email/password or OTP.
- Locks account after 3 wrong attempts.
- Auto logout after inactivity.
- Tracks login history.

7.2 Chef Menu Management

- Add, edit, or delete food items.
- Set prices and availability.
- Upload photos and descriptions.
- View past orders and customer feedback.

7.3 Customer Ordering

- Browse menus by chef or location.
- Add items to cart and checkout.
- Track order status live.
- Rate and review meals.

7.4 Delivery Management

- Assign delivery tasks.
- GPS tracking for delivery routes.
- Notify customer when food arrives.

7.5 Admin Dashboard

- View system stats and analytics.
- Manage users and permissions.
- Handle complaints and feedback.
- Monitor payments and transactions.

7.6 Ratings and Reviews

- Customers can rate meals and leave reviews after each order.
- Helps chefs improve their offerings and builds trust for future customers.
- Admins can monitor feedback to maintain quality across the platform.

8 External Interface Requirements

8.1 User Interfaces

- The system shall provide a responsive design compatible with both web and mobile platforms.
- The interface shall support multilingual accessibility, including Bangla and English.
- Standard UI components shall include:

- Dashboard for user overview
- Menu for browsing food items
- Cart for order management
- Order History for tracking past purchases

8.2 Hardware Interfaces

- The application shall be accessible via smartphones running Android or iOS.
- GPS-enabled devices shall be used for real-time delivery tracking and route optimization.

8.3 Software Interfaces

- The system shall utilize Restful APIs for communication between frontend and backend services.
- Data shall be stored using Firebase or MySQL, depending on deployment configuration.
- Payment processing shall be integrated through Stripe or SSLCOMMERZ.
- Ensuring secure transactions.

8.4 Communications Interfaces

- All data transmission must occur over HTTPS to ensure security and privacy.
- The system shall support push notifications for real-time order updates and alerts.
- Email and SMS services shall be used for confirmations, reminders, and promotional messages.

8.5 Non-Functional Requirements

- **Performance:** The system shall respond to user actions in 2 seconds under normal load.
- **Security:** All user data shall be encrypted and protected using role-based access control.
- **Usability:** The interface must be intuitive and accessible to users with varying technical skills.
- **Reliability:** The system shall maintain 99.9% uptime and include fallback mechanisms for critical services.
- **Scalability:** The architecture shall support future growth in user base and feature expansion.
- **Maintainability:** Code shall be modular and documented to support ongoing updates and debugging.
- **Portability:** The application must run smoothly on all major browsers and mobile operating systems.
- **Compliance:** The system shall adhere to local food safety and data protection regulations.

9 Other Requirements

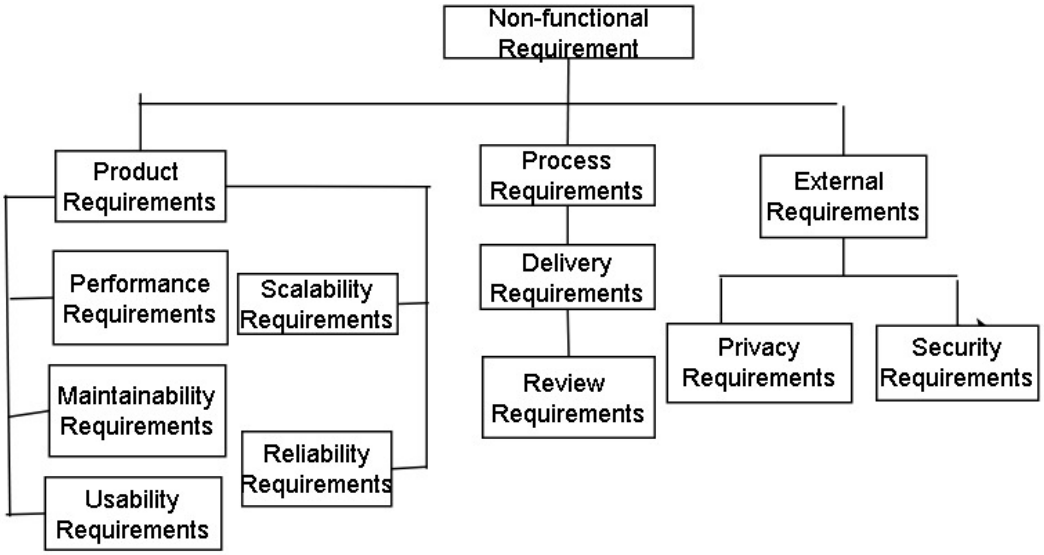

Figure 1: Non-Functional requirements of a System

10 Conclusion

HomelyBites is more than a food delivery app—it's a platform for social impact. By enabling housewives to become entrepreneurs and offering customers healthier alternatives, it bridges a critical gap in the food service ecosystem. With robust technical architecture, thoughtful design, and community-driven goals, ***HomelyBites*** is poised to redefine how we experience homemade food.

PROJECT WORKS	MIZAN(ID-2303005)	ROJONY(ID-2303010)	REDOAN(ID-2303018)
DOCUMENTATION	☐	☑	☑
DIAGRAM / FIGURE	☑	☐	☐

Figure 2: Non-Functional requirements of a System