PickMyDish – Software Design & Modelling

Your Personal Cooking Companion

Kamdeu Yamdjeuson Neil Marshall Tuheu Tchoubi Pempeme
Moussa Fahdil

ICT University
Faculty of Engineering and Technology

December 2025

Project Abstract

- Intelligent recipe recommendation application
- Solves "What should I eat today?" using:
  - Mood-based filtering (7 emotional states)
  - Ingredient availability
  - Time constraints
- Built with **Flutter**, **Node.js**, **MySQL**
- Implements **Layered + Client-Server + Repository** architecture
- **DevOps pipeline** with Jenkins
- 99.8% uptime, 48.5% test coverage

Problem Statement

- **Decision fatigue** in daily meal preparation
- **Resource constraints**: Limited ingredients and cooking time
- **Emotional disconnect**: Traditional apps ignore user's emotional state
- **Accessibility issues**: Lack of personalized, intuitive interfaces

Objectives

Objective	Description
Personalization	Mood-aware recipe filtering (7 emotional states)
Accessibility	Intuitive UI/UX for users of all technical backgrounds
Performance	Fast recipe loading time and offline functionality
Scalability	Architecture supporting 1000+ concurrent users
Reliability	99.8% uptime with robust error handling

Key Features

- Mood-Based Filtering (Happy, Sad, Energetic, Comfort, Healthy, Quick, Light)
- Real-Time Ingredient Matching
- Time-Aware Suggestions
- Personal Favorites
- Recipe Upload System
- Offline Functionality
- User Profiles
- Admin Controls

Target Audience

Persona	Primary Needs	Key Features Used
Cooking Enthusiast	• Mood-based filtering • Recipe organization • Community sharing	• Personalization engine • Recipe upload • Advanced search
Busy Parent	• Quick meal recipes • Family-friendly options • Nutrition tracking	• Time filter (30 mins) • Kid-friendly tags • Meal planning
Student Cook	• Budget-friendly meals • Simple instructions • Beginner guidance	• Step-by-step guide • Ingredient selector • Calorie calculator

Team Roles & Responsibilities

Member	Role	Responsibilities
Kamdeu Yamdjeuson Neil Marshall	CTO / Backend & DevOps Engineer	• DevOps pipeline (Jenkins) • VPS infrastructure (Contabo) • Backend API (Node.js/Express) • MySQL database design • Security & firewall
Tuheu Tchoubi Pempeme Moussa Fahdil	Scrum Master / Frontend & UI/UX Developer	• Sprint planning & standups • Flutter frontend development • UI/UX design & prototyping • State management (Provider) • API integration & testing

Agile Methodology & Scrum

Extreme Programming (XP) Adopted:

- Pair Programming via Discord
- Test-Driven Development (TDD)
- Continuous Integration with Jenkins
- Small Releases (weekly deployment cycles)
- Collective Code Ownership

4-Week Sprint Structure:

- Sprint 1: Foundation (UI, navigation)
- Sprint 2: Core Logic (filtering, database)
- Sprint 3: Features (upload, favorites, profiles)
- Sprint 4: Polish (testing, deployment, docs)

System Architecture

Hybrid Architecture Approach:

- Layered Architecture (separation of concerns)
- Client-Server (distributed access)
- Microservices Elements (independent deployable services)
- Repository Pattern (data access abstraction)

Architectural Layers:

1. Presentation Layer (Flutter UI, Provider)
2. Application Layer (Node.js, Express, Business Logic)
3. Data Access Layer (Repository Pattern, MySQL)
4. Infrastructure Layer (VPS, Nginx, Firewall)

Design Patterns Implementation

Singleton Pattern

- Database Service
- Single connection instance
- Prevents resource conflicts

Factory Pattern

- Model creation
- JSON deserialization
- Flexible object creation

Provider Pattern

- State management
- Observer pattern implementation
- Reactive UI updates

Builder Pattern

- UI widget creation
- Step-by-step construction
- Clean, readable code

DevOps Pipeline

Jenkins CI/CD Pipeline Stages:

1. **Checkout**: Pull latest code from GitHub
2. **Analyze**: Flutter code analysis and linting
3. **Test**: Execute unit and integration tests
4. **Build**: Generate APK and AppBundle
5. **Deploy**: Transfer to VPS and restart services

- Automated testing and deployment
- Hosted on Contabo VPS with Nginx reverse proxy
- SSL/TLS encryption via Let's Encrypt

Results & Performance Metrics

API Performance:

- Avg response time: 85–210ms
- Success rate: 99.5–99.9%
- Uptime: 99.8%

Test Coverage:

- Overall: 84%
- Database Service: 93%
- Recipe Model: 90%
- Screens: 78%

Test Execution:

- Total tests: 31
- Pass rate: 100%
- Unit tests: 15
- Integration tests: 8
- API tests: 5
- Widget tests: 3

Application Screenshots

Registration & Login:

- Clean, intuitive interface
- Password strength indicator
- Guest login option

Home Screen:

- Personalized welcome
- Mood selection (7 emotions)
- Ingredient input
- Time filtering

Recipe Details:

- Complete recipe information
- Ingredients list
- Cooking time and calories
- Favorite toggle

Favorites & Profile:

- User's favorite recipes
- Profile information
- Member since date
- Logout functionality

Challenges & Solutions

Challenge	**Solution**
VPS Configuration Complexity	Studied Contabo/Nginx documentation, configured UFW firewall, optimized Nginx settings
State Management in Flutter	Implemented Provider pattern, created custom ChangeNotifiers, added state persistence
Database Optimization	Normalized database schema, added composite indexes, implemented connection pooling
CI/CD Pipeline Automation	Studied Jenkins documentation, added debugging outputs, created automated scripts

Future Enhancements

Priority	Feature
High	• AI Recipe Recommendations (machine learning) • Meal Planning Calendar with grocery lists
Medium	• Social Features (profiles, following, sharing) • Multi-language Support (internationalization) • Advanced Analytics (user behavior insights)
Low	• Voice Commands (voice-controlled navigation) • AR Cooking Assistance (augmented reality guidance)

Lessons Learned

Technical Lessons:

- Importance of proper database indexing
- Value of comprehensive error handling
- Benefits of automated testing in CI/CD
- Necessity of proper logging and monitoring
- Advantages of containerization for consistency

Project Management Lessons:

- Daily standups prevent misalignment
- Kanban boards improve workflow visibility
- Pair programming enhances code quality
- Regular deployments reduce integration risks
- Documentation saves time in the long term

Conclusion

- Successfully delivered a fully functional recipe recommendation application
- Demonstrated mastery of software architecture patterns
- Implemented 4+ design patterns following SOLID principles
- Established complete DevOps pipeline with Jenkins
- Achieved professional deployment on production-ready VPS
- Maintained comprehensive documentation including UML diagrams

- **GitHub Repository:**
  `https://github.com/Kymmarshall/Pick-My-Dish`
  `https://github.com/Kymmarshall/Pick-My-Dish-Backend`
- **Live Application:**
  `pickmydish.duckdns.org`

Thank You!

Questions?

Instructor:	Eng. Tekoh Palma Achu
Course:	SEN3140 - Software Design and Modelling
Date:	December 2025
University:	ICT University - Faculty of Engineering and Technology