



CS5002NI Software Engineering

35% Weightage Coursework AY 2025- 26
Credit: 30

Student Name: Abiram Bhatta

London Met ID: 24046558

College ID: NP01A14A20144

Assignment Due Date: April 30, 2026 (6:00pm)

Assignment Submission Date: April 30, 2026 (6:00pm)

Word Count: 4683

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

1. Introduction	5
1.1 Background	5
1.2 Aims and Objectives	5
1.3 Scope of the Report	5
2 Software Architecture Diagram	6
2.1 Question 1 - Work Breakdown Structure and Gantt Chart	6
2.2 WBS Explanation	7
2.3 Gantt Chart	8
3. Question 2 - Use Case Model	9
3.1 Use Case Diagram	9
3.2 High-Level Use Case Descriptions	10
3.3 Expanded Use Case Description	11
3.3.1 Use Case 3: Book Table	11
3.3.2 Use Case 6: Place Order	13
4. Question 3 – Sequence Diagram	15
5. Question 4 - Activity Diagram	16
6. Question 5 - Domain Classes and Analysis Class Diagram	17
7. Question 6 - Further Development Plan	19
7.1 Selected Development Methodology: DSDM	19
7.2 Architectural Choice	19
7.3 Design Patterns	20
7.4 Development Plan, Tools and Priority Order	20
7.5 Testing Plan: STLC and Test Types	22
7.5.1 Black-Box Testing	23
7.5.2 White-Box Testing	23
7.5.3 Test Levels and Test Types	24
7.5.4 Feature-Based Test Matrix	25
7.6 Maintenance Plan	25
8. Question 7 - Design Prototypes	26

8.1 Prototype 1.....	27
8.2 Prototype 2.....	27
8.3 Prototype 3.....	28
8.4 Prototype 4.....	28
8.5 Prototype 5.....	29
8.6 Prototype 6.....	29
8.7 Prototype 7.....	30
8.8 Prototype 8.....	30
8.9 Prototype 9.....	31
8.10 Prototype 10.....	32
8.11 Prototype 11.....	32
8.12 Prototype 12.....	33
8.13 Prototype 13.....	34
8.14 Prototype 14.....	34
8.15 Prototype 15.....	34
9. Conclusion	35
10. References.....	35

Table of Figures:

Figure 1: DSDM Work Breakdown Structure for the Gokyo Bistro Restaurant Management System	6
Figure 2: Gyannt chart following the wbs to complete the projec	8
Figure 3: Use Case Diagram for the Gokyo Bistro Restaurant Management System	9
Figure 4: Sequence Diagram - UC3 Book Table	15
Figure 5: Activity Diagram - UC6 Place Order.....	16
Figure 6: Analysis Class Diagram with inheritance, aggregation, composition and multiplicities	18
Figure 7: Proposed Layered MVC Architecture	19
Figure 8: STLC flow adapted for DSDM timeboxes.....	22
Figure 9: Customer landing page with reservation, order and offer entry points.	27
Figure 10: Login and registration screen for member and admin access.	27
Figure 11: Member dashboard showing bookings, orders, receipts and feedback actions.	28
Figure 12: Table availability search with date, time, party size and result status.	28
Figure 13: Interactive floor-plan selection with indoor/outdoor table availability.	29
Figure 14: Reservation details and optional beverage choices including premium reserve	29
Figure 15: Rs. 1200 advance payment screen with clear cancellation policy	30
Figure 16: Booking confirmation and cancellation/refund preview	31
Figure 17: Menu browsing and item selection for members.	32
Figure 18: Home delivery tracking screen after order confirmation.	33
Figure 19: Bill and receipt generation screen with admin billing controls.	33
Figure 20: Rating and feedback screen for food, staff, ambience and delivery.	34
Figure 21: Admin financial report dashboard with revenue prediction and recommendations..	34
Figure 22: Admin menu CRUD, offers and discount scheme management.	35

1. Introduction

1.1 Background

Gokyo Bistro is a restaurant in Jhamsikhel, Lalitpur that needs a digital system to manage table reservations, food and beverage ordering, home delivery, payments, offers, feedback, and administrative reporting. The coursework brief states that the restaurant is facing table arrangement and ordering problems because customer demand has increased, some customers return due to full occupancy, and more customers now ask for home delivery. The proposed Restaurant Management System therefore needs to reduce manual errors, avoid overbooking, improve communication between customers and staff, and give the admin team better visibility of revenue and menu performance.

This report applies Object-Oriented Analysis and Design (OOAD) to the proposed system. It includes a DSDM-based project structure, use case model, expanded use case descriptions, behaviour diagrams, an analysis class model, a further development plan, a testing strategy and a set of user interface prototypes.

1.2 Aims and Objectives

- Identify the functional scope of the Gokyo Bistro Restaurant Management System using WBS and use case modelling.
- Model system behaviour for table booking and order placement using sequence and activity diagrams.
- Identify the main domain classes and relationships needed for a future object-oriented implementation.
- Explain how the project should proceed after analysis and design using a suitable development methodology.
- Plan architecture, design patterns, development priorities, testing and maintenance activities.
- Prepare high-fidelity UI prototypes for the major features described in the coursework specification.'

1.3 Scope of the Report

The report focuses on the software engineering analysis and planning deliverables requested in the coursework. It does not implement the system in code, but it provides the design and development basis for implementation. The system scope includes registration, login, table booking, beverage choice selection, advance payment, cancellation/refund handling, order placement, home delivery, bill/receipt generation, menu management, experience rating, offers/schemes, financial reporting, revenue prediction and menu recommendation.

2 Software Architecture Diagram

Methodology Alignment: DSDM

The selected methodology for this coursework is DSDM (Dynamic Systems Development Method), also known in modern guidance as the DSDM Agile Project Framework. DSDM is suitable for this project because Gokyo Bistro has a clearly defined business need, but the detailed solution can be refined through prototypes, stakeholder feedback and timeboxed delivery. DSDM is iterative and incremental, fixes time and quality, and uses MoSCoW prioritisation to flex lower-priority scope when necessary.

2.1 Question 1 - Work Breakdown Structure and Gantt Chart

The WBS below is structured around the DSDM phases: Pre-Project, Feasibility, Foundations, Evolutionary Development, Deployment and Post-Project. Evolutionary Development is further broken into three practical timeboxes for this restaurant system: access/menu/reservation; payments/orders/billing; and admin/analytics/engagement.

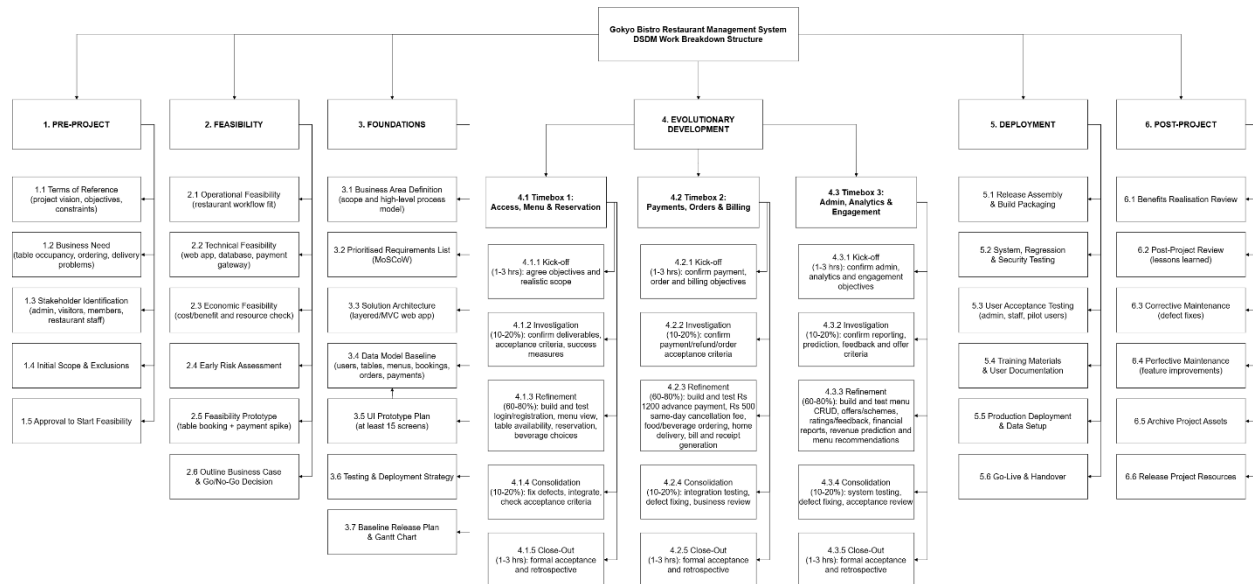


Figure 1: DSDM Work Breakdown Structure for the Gokyo Bistro Restaurant Management System

2.2 WBS Explanation

DSDM Phase	Main Purpose in the Gokyo Bistro Project
1. Pre-Project	Confirm project vision, business problem, stakeholders, initial scope and approval to investigate further.
2. Feasibility	Check operational, technical and economic feasibility, identify early risks and create a small feasibility prototype for booking/payment.
3. Foundations	Define the business area, high-level requirements, architecture, data model, prototype plan, testing/deployment strategy and baseline delivery plan.
4. Evolutionary Development	Develop the solution in DSDM timeboxes with kick-off, investigation, refinement, consolidation and close-out activities.
5. Deployment	Assemble the release, complete system/regression/security testing, run UAT, prepare training and deploy to production.
6. Post-Project	Review benefits, capture lessons learned, perform maintenance and archive project assets.

2.3 Gantt Chart

The Gantt chart follows the DSDM methodology selected for this project. Therefore, the chart is structured around DSDM phases such as Pre-Project, Feasibility, Foundations, Evolutionary Development, Deployment, and Post-Project. The development work is divided into DSDM timeboxes rather than Scrum sprints. Each timebox includes activities such as kick-off, investigation, refinement, consolidation, testing, and close-out. This ensures that the schedule remains consistent with the DSDM WBS and the overall project methodology.

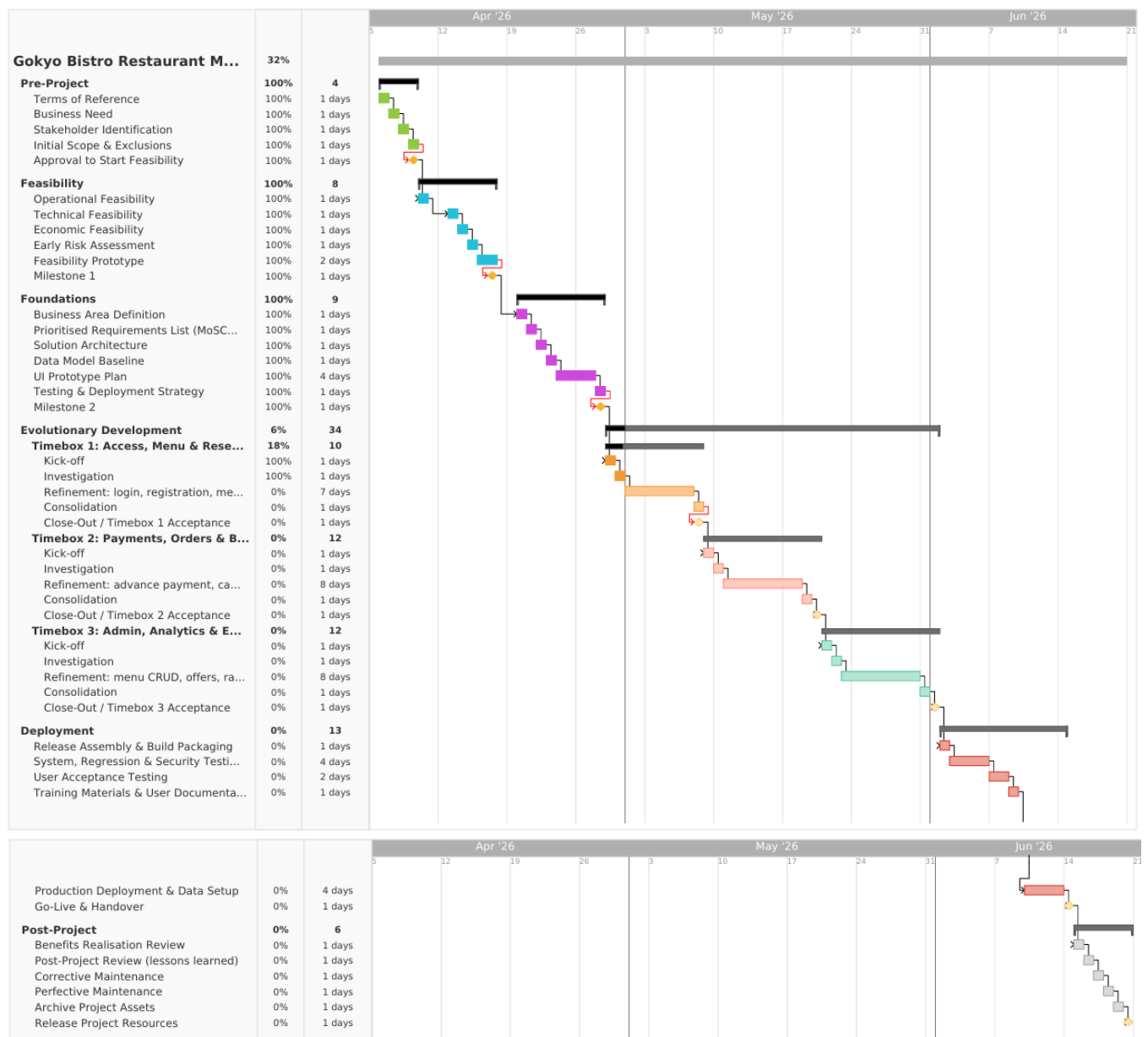


Figure 2: Gyannt chart following the wbs to complete the projec

3. Question 2 - Use Case Model

3.1 Use Case Diagram

The use case model identifies the main system actors and the services they expect from the system. The primary actors are Visitor, Member and Admin. Visitors may browse, register, check availability and book as a guest by providing contact details. Members can log in, book tables, choose beverages, place orders and request home delivery. Admin users maintain menu items and offers, generate bills and produce financial reports.

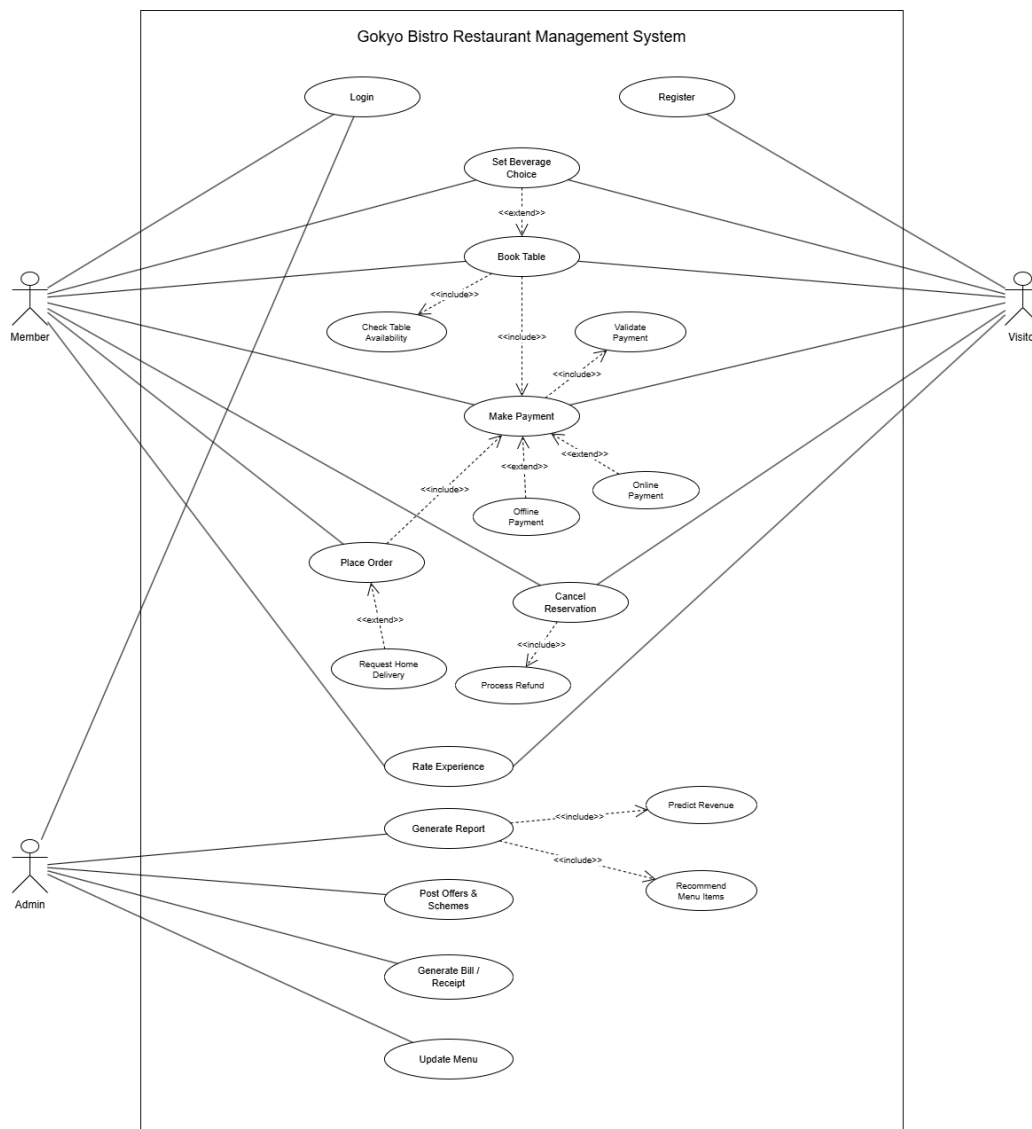


Figure 3: Use Case Diagram for the Gokyo Bistro Restaurant Management System

3.2 High-Level Use Case Descriptions

ID	Use Case	Actors	Description	Precondition	Postcondition
UC1	Register	Visitor	Creates a member account using name, email, phone, password and membership type.	Visitor is not already registered.	Member account is created.
UC2	Login	Admin, Member	Authenticates the user and starts a secure session.	User has valid credentials.	User enters the appropriate dashboard.
UC3	Book Table	Visitor, Member	Checks table availability and reserves a table for a selected date and time.	User has entered booking details and table availability data exists.	Reservation is created after successful advance payment.
UC4	Set Beverage Choices	Visitor, Member, Premium Member	Stores expected beverage preferences during booking; premium members can reserve wine/whiskey choices.	Booking is being created.	Beverage choices are attached to the reservation.
UC5	Make Payment	Visitor, Member	Pays Rs. 1200 advance for table booking or pays for an order.	Payable transaction exists.	Payment status is recorded.
UC6	Place Order	Member	Allows a member to select food and beverages and confirm an order.	Member is logged in and menu is available.	Order is created and sent to kitchen/bar.
UC7	Request Home Delivery	Member	Adds delivery address and delivery charge to an order.	Member is placing an order.	Delivery is scheduled.
UC8	Cancel Reservation	Visitor, Member	Cancels a booking and applies the refund policy.	Reservation exists and is cancellable.	Reservation status and refund amount are updated.
UC9	Rate Experience	Visitor, Member	Submits scores for food, staff behaviour, ambience and	Customer has completed dining or delivery	Rating and feedback are stored.

			written feedback.	experience.	
UC10	Generate Report	Admin	Generates financial report, revenue prediction and menu recommendations.	Admin is logged in.	Report is generated/exported.
UC11	Generate Bill/Receipt	Admin	Creates bills and receipts for orders or bookings.	Order or booking payment exists.	Bill or receipt is generated.
UC12	Update Menu	Admin	Adds, edits, deletes and updates menu items.	Admin is logged in.	Menu information is updated.
UC13	Post Offers and Schemes	Admin	Creates and publishes special offers and discount schemes.	Admin is logged in.	Offer or scheme is published to users.

3.3 Expanded Use Case Description

3.3.1 Use Case 3: Book Table

Actors:

- Visitor
- **Member**

Preconditions:

- The user must have access to the system.
- The table availability information must be up to date.
- The user must provide a preferred booking date and time.

Main Flow:

1. The user opens the Book Table option from the system menu.
2. The system presents a reservation form containing the required booking details, including:
 - Date
 - Time
 - Duration
 - Number of guests
 - Seating or table preference
3. The user enters the required reservation information.
4. The system searches the table records to check whether any suitable tables are available for the selected date and time.
5. If suitable tables are available:

- The system shows a list of available tables.
 - The user chooses the preferred table.
 - The system displays the advance booking amount of Rs. 1200.
 - The user continues to the payment step.
6. If no tables are available:
- The system provides alternative time slots or dates.
 - The user may either choose one of the suggested alternatives or exit the booking process.

Alternative Flows:

A1: User Cancels the Reservation

- The user selects the Cancel option.
- The system stops the booking process and redirects the user to the main menu.

A2: User Changes Booking Information

- The user edits the selected date, time, or table preference.
- The system checks table availability again using the updated details.
- The process continues from the availability checking step.

Postconditions:

- A table reservation is created for the user.
- The booking details are stored in the system database.
- The user receives confirmation of the reservation.
- The advance payment of Rs. 1200 is recorded against the booking.

Special Requirements:

- Premium members are allowed to reserve wine or whiskey preferences along with their table booking.
- The cancellation policy must be applied as follows:
 - Cancellation before the final booking day results in a full refund.
 - Cancellation on the final booking day results in a Rs. 500 deduction from the advance payment.

3.3.2 Use Case 6: Place Order

Actor:

- Member

Preconditions:

- The user must be registered as a member and logged into the system.
- The member must have access to a valid payment option.
- Menu items must be available in the system.

Main Flow:

1. The member selects the Place Order option from the system.
2. The system displays the available menu sections, such as:
 - Food
 - Beverages
3. The member chooses the required menu items and enters the quantity for each item.
4. The system calculates the subtotal based on the selected items.
5. The system checks and displays any available offers or discounts.
6. The member applies a discount if it is suitable for the order.
7. The system recalculates and displays the final payable amount.
8. The member selects a payment method, such as:
 - Online payment
 - Cash on delivery, if home delivery is selected
9. If online payment is selected:
 - The system begins the online payment process.
 - The member completes the payment.
 - The system verifies and confirms the payment status.
10. The system generates an order confirmation.
11. The confirmed order is forwarded to the kitchen or bar for preparation.

Alternative Flows:

A1: Home Delivery Selected

- The system displays a delivery address form.
- The member enters a new address or confirms a saved address.
- The system adds the delivery charge to the order total.
- The process continues from the final amount calculation step.

A2: Payment Failure

- The system displays a payment failure message.
- The member is given the option to retry the payment or cancel the order.

- If the member cancels the order, the system discards the order request.

Postconditions:

- An order record is created in the database.
- Payment is successfully recorded where required.
- The member receives order confirmation.
- The order is sent to the kitchen or bar for preparation.
- If home delivery is selected, the delivery request is scheduled.

Special Requirements:

- Only registered members are permitted to place orders.
- Online orders must be confirmed only after successful payment.
- Home delivery is available only to members.
- The system must maintain order history for each member.

4. Question 3 – Sequence Diagram

The sequence diagram models the UC3 Book Table use case. It shows the interaction between the Visitor/Member, BookTable UI, BookTableController, Table, Payment and Reservation objects. The most important logic is that the system must check availability before payment, then create the reservation only after successful payment.

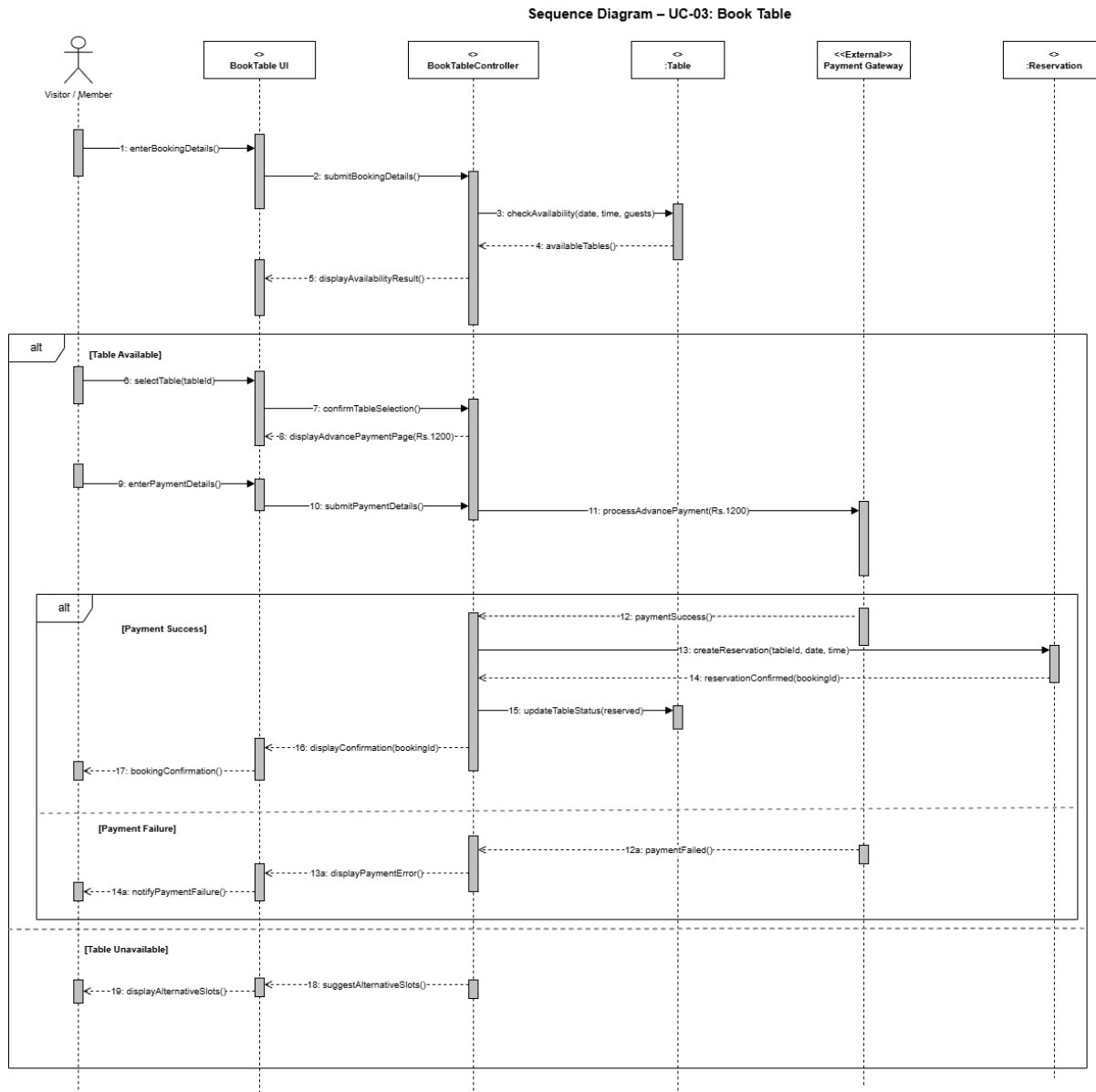


Figure 4: Sequence Diagram - UC3 Book Table

5. Question 4 - Activity Diagram

The activity diagram models the UC6 Place Order use case, which was not used for the sequence diagram. It highlights menu browsing, cart review, dine-in versus home delivery decision, delivery charge calculation, payment processing and order confirmation.

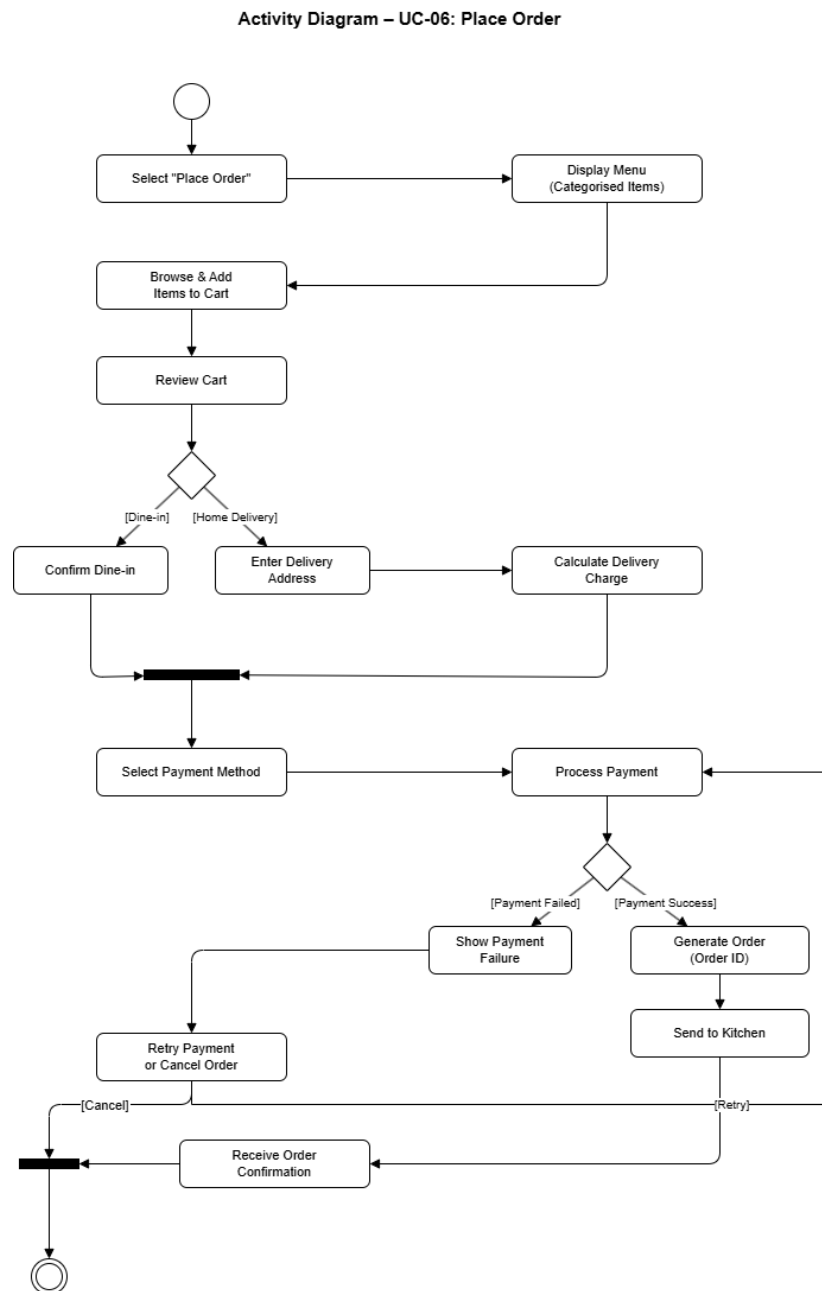


Figure 5: Activity Diagram - UC6 Place Order

6. Question 5 - Domain Classes and Analysis Class Diagram

The class diagram is based on the main use cases of the Gokyo Bistro Restaurant Management System. It shows the key classes such as users, reservations, orders, payments, menu items, feedback, reports, and offers. The class diagram shows the main classes of the Gokyo Bistro Restaurant Management System and how they are connected. It includes key classes such as users, reservations, orders, payments, menu items, feedback, reports, and offers. The diagram also shows relationships such as inheritance, association, aggregation, composition, and multiplicities to explain the structure of the system clearly.

Domain Class	Purpose	Related Use Case(s)
User	Base actor entity for authentication and profile details.	Login, registration, rating, booking.
Member	Registered customer who can place orders and view history.	Login, book table, place order, delivery, rating.
Admin	Restaurant administrator who manages menu, reports, bills and offers.	Update menu, post offers, generate bills/reports.
Visitor	Guest user who can browse and book as guest.	Register, book table, rate experience.
Reservation	Booking record for date/time, guests and table status.	Book table, cancel reservation.
Table	Restaurant table with capacity, location and availability state.	Check availability, reserve table.
BeverageChoice	Optional expected beverage choices attached to reservation.	Set beverage choices.
Payment	Payment/refund transaction for booking or order.	Make payment, refund cancellation.
Bill	Bill/receipt issued for order or booking payment.	Generate bill/receipt.
Order	Food and beverage order placed by member.	Place order.
OrderItem	Line item with quantity and subtotal.	Place order, bill generation.
Delivery	Home delivery information for an order.	Request home delivery.
Menu	Collection of menu items.	Browse menu, update menu.
MenuItem	Food or beverage item with price/category.	Place order, update menu.
Offer	Discount or special scheme.	Post offers, apply

7. Question 6 - Further Development Plan

7.1 Selected Development Methodology: DSDM

DSDM is selected because it supports the whole project lifecycle, not only coding. The method begins with enough discovery and foundation work before development, then evolves the solution iteratively through timeboxes. This fits the Gokyo Bistro project because the core business needs are known, but UI details, reporting expectations and integration details can be refined through prototypes and user feedback. DSDM also supports modelling and prototyping during Feasibility and Foundations, and uses timeboxes and MoSCoW prioritisation to protect deadlines while maintaining quality.

- Pre-Project and Feasibility confirm that the restaurant problem is worth solving and technically achievable.
- Foundations define the high-level architecture, prioritised requirements, UI prototypes, test strategy and release plan.
- Evolutionary Development delivers the solution in three timeboxes, with testing included inside development rather than left until the end.
- Deployment assembles the release, completes regression/security/UAT checks and hands the system to users.
- Post-Project reviews realised benefits and starts structured maintenance.

7.2 Architectural Choice

A layered MVC web application architecture is proposed. The View layer handles web pages and UI prototypes; Controllers receive user requests; Services implement business rules such as reservation, refund and ordering; Repositories isolate database access; and external services such as payment gateway and notifications are accessed through adapters. This architecture is easy to test because business logic can be unit-tested in services, database operations can be mocked through repositories, and user flows can be tested end-to-end through the presentation layer.

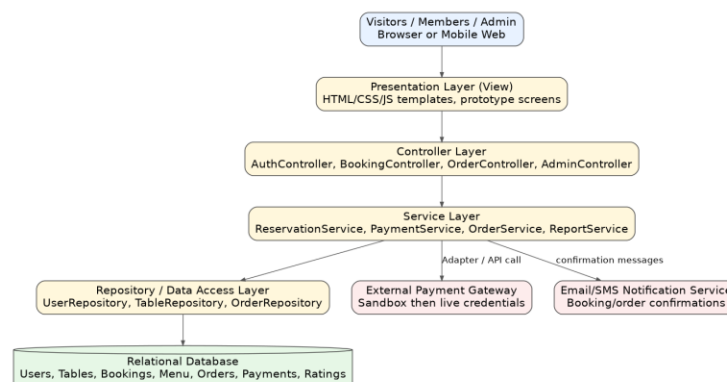


Figure 7: Proposed Layered MVC Architecture

Layer	Responsibility
Presentation/View	HTML/CSS/JavaScript or template views for booking, menu, cart, reports and admin screens.
Controller	Routes requests such as login, book table, place order, cancel booking and generate report.
Service	Contains business logic: availability checking, payment calculation, refund rules, order totals, revenue prediction.
Repository/Data Access	Reads and writes users, tables, reservations, menu items, orders, payments and ratings.
External Integrations	Payment gateway, email/SMS notification and possible analytics/export libraries.

7.3 Design Patterns

Pattern	How it applies to Gokyo Bistro
MVC	Separates user interface, control logic and data/business model. It supports clean screens and manageable controller code.
Repository Pattern	Keeps database code separate from services, making unit testing and later database changes easier.
Service Layer	Centralises business rules such as Rs. 1200 advance, Rs. 500 same-day cancellation deduction and order total calculation.
Strategy Pattern	Allows different payment methods or discount calculation rules to be swapped without rewriting order/booking code.
Adapter Pattern	Wraps payment gateway and notification APIs so the internal system is not tightly coupled to external providers.
Factory Method	Creates different report types such as daily sales, monthly finance, revenue prediction and menu recommendation.

7.4 Development Plan, Tools and Priority Order

The project should be delivered in DSDM timeboxes. Requirements are prioritised using MoSCoW: Must Have items are essential for a usable first release, Should Have items add significant value, Could Have items can be included if time allows, and Won't Have items are deferred. DSDM guidance recommends using MoSCoW to manage business expectations and protect deadline confidence.

Area	Suggested Tool/Resource
Frontend	HTML5, CSS3, JavaScript/TypeScript; responsive design for browser/mobile use.
Backend	Java/Spring Boot, Python/Django or Node/Express. Choose one platform and keep it consistent.
Database	MySQL/PostgreSQL for users, bookings, tables, menu, orders, payments and ratings.
UML/Design	draw.io/diagrams.net for UML diagrams; Figma/Balsamiq/draw.io for UI prototypes.
Testing	JUnit/PyTest/Jest for unit tests; Postman for API tests; Selenium/Cypress/Playwright for end-to-end regression.
Project/Version Control	Git/GitHub or GitLab, issue tracker, Kanban board, test evidence folder.
Deployment	Staging server, production server, environment variables for payment credentials, backup strategy.

Increment/Feature Group	MoSCoW Priority	Planned Deliverables
Timebox 1 - Access, Menu and Reservation	Must Have	Registration, login, menu view, table availability, reservation, optional beverage choices.
Timebox 2 - Payments, Orders and Billing	Must Have	Rs. 1200 advance payment, refund/cancellation policy, place order, home delivery, bill/receipt generation.
Timebox 3 - Admin, Analytics and Engagement	Should Have	Admin menu CRUD, offers, ratings, financial reports, revenue prediction and recommendations.
Enhancement backlog	Could Have	Email/SMS reminders, loyalty points, advanced recommendation model, table layout drag/drop.
Deferred	Won't Have this time	Native mobile app, multi-branch restaurant management, full ERP/accounting integration.

7.5 Testing Plan: STLC and Test Types

Testing should begin at the Foundations stage, not only after coding. This follows a shift-left approach: acceptance criteria, test conditions and sample test data are prepared from the use cases before development starts. ISTQB describes the test process as including planning, monitoring/control, analysis, design, implementation, execution and completion, and notes that in Agile development these activities often happen iteratively and continuously. Therefore, each DSDM timebox should include test design, execution, defect fixing and regression before close-out (ISTQB, 2021).

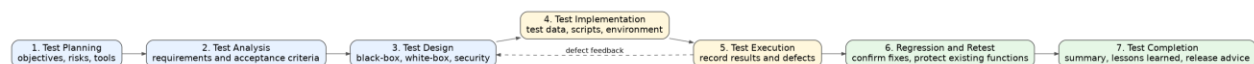


Figure 8: STLC flow adapted for DSDM timeboxes

STLC Activity	Application to the Project
Test Planning	Define scope, risks, entry/exit criteria, test levels, tools, responsibilities and schedule for each timebox.
Test Analysis	Study use cases, class diagram, UI prototypes and acceptance criteria to identify what must be tested.
Test Design	Create black-box tests from specifications and white-box tests from code/design logic.
Test Implementation	Prepare test data, scripts, mock payment gateway, database seed records and test environment.
Test Execution	Run manual and automated tests, record actual results and log defects.
Regression and Retest	Retest fixed defects and rerun a regression pack to ensure existing features still work.
Test Completion	Produce test summary, unresolved defect list, lessons learned and release recommendation.

7.5.1 Black-Box Testing

Black-box testing checks the system from the outside using requirements, inputs and expected outputs. It is appropriate for booking forms, payment paths, order totals, offers, delivery choices and admin screens because testers do not need to inspect the source code to validate those behaviours.

Technique	Example for Gokyo Bistro
Equivalence Partitioning	Guest count: valid values such as 1-8 and invalid values such as 0, negative numbers or very large groups.
Boundary Value Analysis	Cancellation date: one day before booking versus same day to verify full refund versus Rs. 500 deduction.
Decision Table Testing	Payment outcome combined with table availability: available + paid = confirmed; available + failed = not confirmed; unavailable = alternative suggested.
State Transition Testing	Reservation states: Pending -> Confirmed -> Cancelled/Completed; Order states: Cart -> Pending Payment -> Confirmed -> Preparing -> Delivered.
Use Case Testing	End-to-end Book Table and Place Order scenarios from the expanded use case descriptions.

7.5.2 White-Box Testing

White-box testing uses knowledge of the internal structure of components. Developers should apply it mainly at unit level for service and utility classes. For example, `ReservationService.calculateRefund()` should be tested for branches where cancellation is before the final day, on the final day, already cancelled or payment not found. `PaymentService` should be tested with mocked gateway responses, and `ReportService` should be tested with controlled sales data.

White-Box Technique	Example
Statement/Branch Coverage	Ensure all branches in refund calculation, offer application and payment status handling are executed.
Path Testing	Cover success and failure paths in booking creation and order confirmation.
Data Flow Testing	Check that totals, discounts, delivery charges and refund amounts are calculated and stored correctly.
Code Review / Static Checks	Review authentication, validation, SQL parameterisation and error handling before deployment.

7.5.3 Test Levels and Test Types

Test Level/Type	Purpose	Approach	Example
Unit Testing	Individual services and model methods.	White-box	Reservation refund, order total, discount calculation, report calculation.
Integration Testing	Interaction between modules.	Black-box + API	Booking with payment gateway stub, order with delivery, bill linked to payment.
System Testing	Complete application flow.	Black-box	Visitor/member books table, pays advance, receives confirmation; member places delivery order.
User Acceptance Testing	Business validation by admin/staff/pilot users.	Acceptance	Admin confirms reports and menu management match restaurant workflow.
Regression Testing	Previously working features after change/fix.	Automated + manual	After modifying payment logic, rerun booking, cancellation, order and bill tests.
Smoke Testing	Build verification after deployment.	Black-box	Login works, menu loads, booking search opens, admin dashboard opens.
Security Testing	Authentication, authorisation, session and input validation.	Security	Admin-only routes blocked for members; secure session cookies; SQL injection checks.
Performance Testing	Response time under load.	Non-functional	Table search and menu browsing during peak dinner traffic.
Usability Testing	Ease of use for customers and admin.	UX/UAT	Users can complete booking/payment without confusion.
Compatibility Testing	Different browsers/devices.	Non-functional	Chrome/Edge/Firefox and mobile screen sizes.

7.5.4 Feature-Based Test Matrix

Feature	Representative Test Cases	Testing Used
Login/Registration	Valid login, invalid password, duplicate email, weak password, admin/member role access.	Unit, system, security, regression.
Book Table	Available table, unavailable table, guest count boundaries, premium beverage, payment success/failure.	Black-box, integration, regression.
Cancellation/Refund	Before final day full refund; same-day Rs. 500 deduction; already cancelled booking.	Boundary, decision table, unit, regression.
Place Order	Add/remove items, out-of-stock item, offer discount, online payment, COD delivery.	Use case, integration, regression.
Admin Menu/Offers	Add, update, delete menu item; create expired/valid offer; validate discount limits.	System, role-based security.
Reports/Prediction	Correct totals for date range, recommendation output, export report.	Data-driven, system, UAT.
Rating/Feedback	Valid scores 1-5, invalid score blocked, feedback stored against user/order/booking.	Boundary, system, regression.

7.6 Maintenance Plan

Maintenance starts after deployment and continues during Post-Project. It should be planned as part of the DSDM handover so the restaurant can safely operate the system after release.

Maintenance Type	Planned Activities
Corrective Maintenance	Fix production defects such as failed receipt generation or incorrect refund amounts.
Adaptive Maintenance	Update payment gateway, browser compatibility, hosting environment or legal/tax requirements.
Perfective Maintenance	Improve features such as better revenue prediction, loyalty scheme or enhanced recommendation engine.
Preventive Maintenance	Refactor code, update dependencies, improve logs, strengthen security and optimise database queries.
Operational Support	Daily backups, monitoring, incident log, admin training and user documentation.
Release Control	Use version numbers, changelog, staging deployment and regression suite before production release.

8. Question 7 - Design Prototypes

The coursework requires at least **15 separate UI prototypes** that cover the main features of the system. The prototypes for Gokyo Bistro have been designed as clean landscape mock-ups, showing key screens such as registration, login, table booking, ordering, payment, delivery, feedback, reports, and menu management. Each screen uses clear spacing, readable labels, and properly aligned buttons, fields, cards, and tables. The top navigation bar also separates the **Gokyo Bistro** brand name from the **Restaurant Management System** subtitle for a cleaner and more professional layout.

Prototype	Feature Covered
1	Customer landing page with reservation, order and offer entry points.
2	Login and registration screen for member and admin access.
3	Member dashboard showing bookings, orders, receipts and feedback actions.
4	Table availability search with date, time, party size and result status.
5	Interactive floor-plan selection with indoor/outdoor table availability.
6	Reservation details and optional beverage choices including premium reserve.
7	Rs. 1200 advance payment screen with clear cancellation policy.
8	Booking confirmation and cancellation/refund preview.
9	Menu browsing and item selection for members.
10	Order basket, offer application, delivery option and checkout.
11	Home delivery tracking screen after order confirmation.
12	Bill and receipt generation screen with admin billing controls.
13	Rating and feedback screen for food, staff, ambience and delivery.
14	Admin financial report dashboard with revenue prediction and recommendations.
15	Admin menu CRUD, offers and discount scheme management.

8.1 Prototype 1

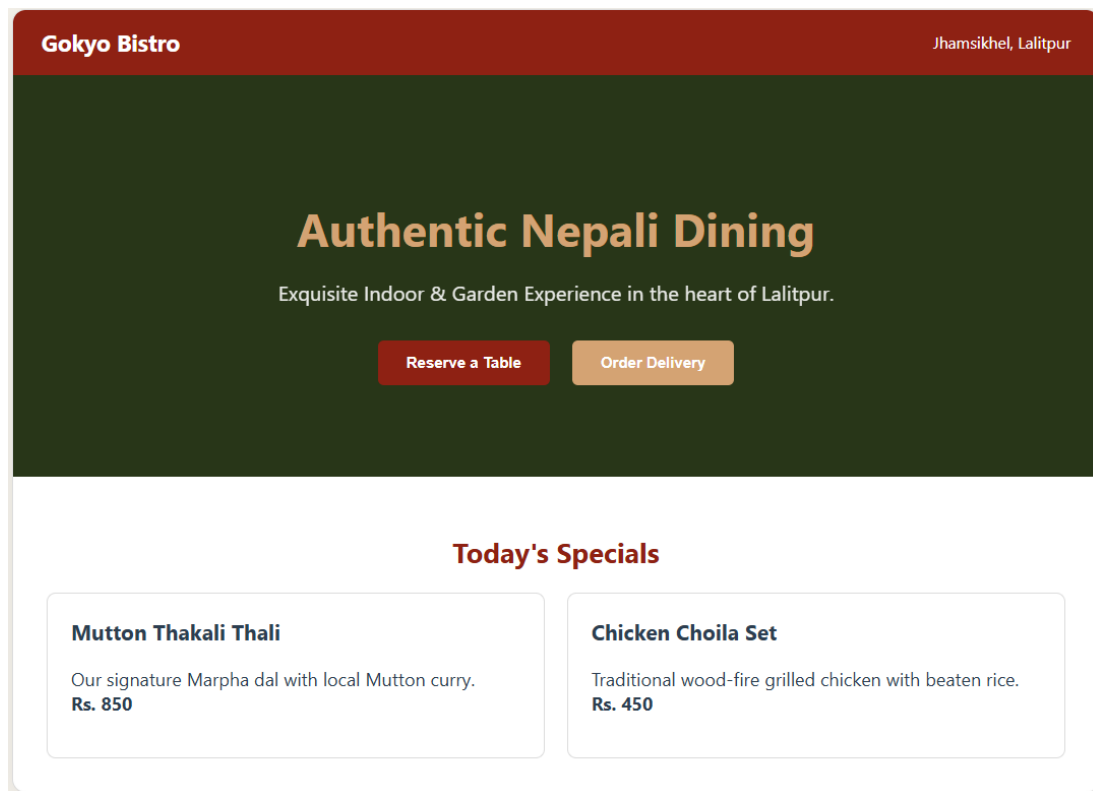


Figure 9: Customer landing page with reservation, order and offer entry points.

8.2 Prototype 2

The image shows a web page for 'Member Access'. The header is dark red with 'Member Access' on the left. The main content area is white. There are two tabs: 'Login' (active) and 'Register'. Below the tabs are two input fields: 'Email / Phone' with the value 'binod@email.com' and 'Password' with the value '*****'. There is a checkbox for 'Remember me' and a link for 'Forgot Password?'. Below these is a dark red button labeled 'Enter Bistro Portal'. At the bottom, there is a link for 'Staff or Admin? Login Here'.

Figure 10: Login and registration screen for member and admin access.

8.3 Prototype 3

Dashboard

Namaste, Binod Chaudhary

Reward Points

1,450

Silver Member Status

Quick Actions

View Reservations

Order History

Recent Bookings

Table #04 (Garden Area)

Oct 30, 2025 • 7:30 PM • 4 People

UPCOMING

Modify

Figure 11: Member dashboard showing bookings, orders, receipts and feedback actions.

8.4 Prototype 4

Find a Table

Book Your Experience

Reservation Date

30-Oct-2025

Arrival Time

7:30 PM

No. of Guests

4 People

Search Available Tables

Figure 12: Table availability search with date, time, party size and result status.

8.5 Prototype 5

Choose Your Spot

Oct 30, 7:30 PM

Interactive Floor Plan

● Available ● Booked

INDOOR DINING

T1

T2

T3

T4

GARDEN AREA (OUTDOOR)

O1

O2

O3

O4

O5

Selected: Garden Table #O4 (Premium View)

Confirm Selection

Figure 13: Interactive floor-plan selection with indoor/outdoor table availability.

8.6 Prototype 6

Set Beverage Choices

Welcome Drinks (Optional)

Pre-order drinks so they are ready when you arrive.

Standard Refreshments

Choice 1

Lassi (Sweet/Salted)

Premium Reserve [Members Only]

Exclusive for Members.

Select Bottle

Old Durbar Black Chimney

Skip

Save & Pay Advance

Figure 14: Reservation details and optional beverage choices including premium reserve

8.7 Prototype 7

Advance Payment

Amount to Pay: Rs. 1200

This amount is required to secure your reservation at Gokyo Bistro.

Cancellation Policy

- **Before the Day:** 100% Refundable.
- **On the Same Day:** Rs. 500 will be deducted as a cancellation fee.

Select Payment Method

eSewa / Khalti

FonePay

Card

Pay Rs. 1200 & Confirm

Figure 15: Rs. 1200 advance payment screen with clear cancellation policy

8.8 Prototype 8

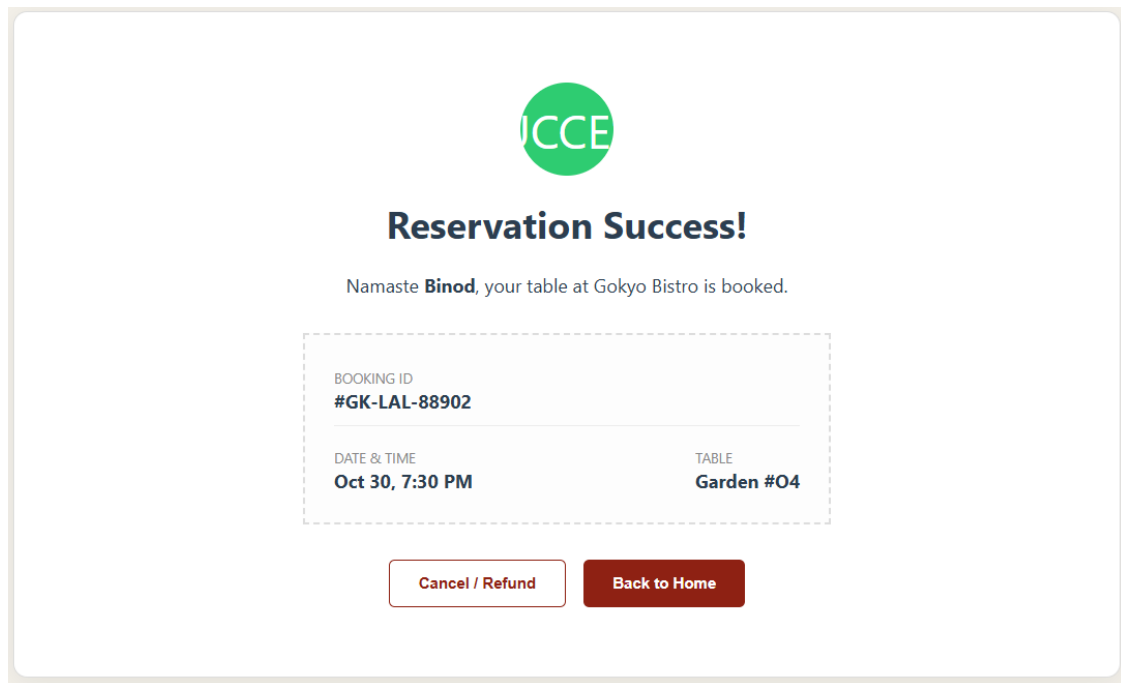


Figure 16: Booking confirmation and cancellation/refund preview

8.9 Prototype 9

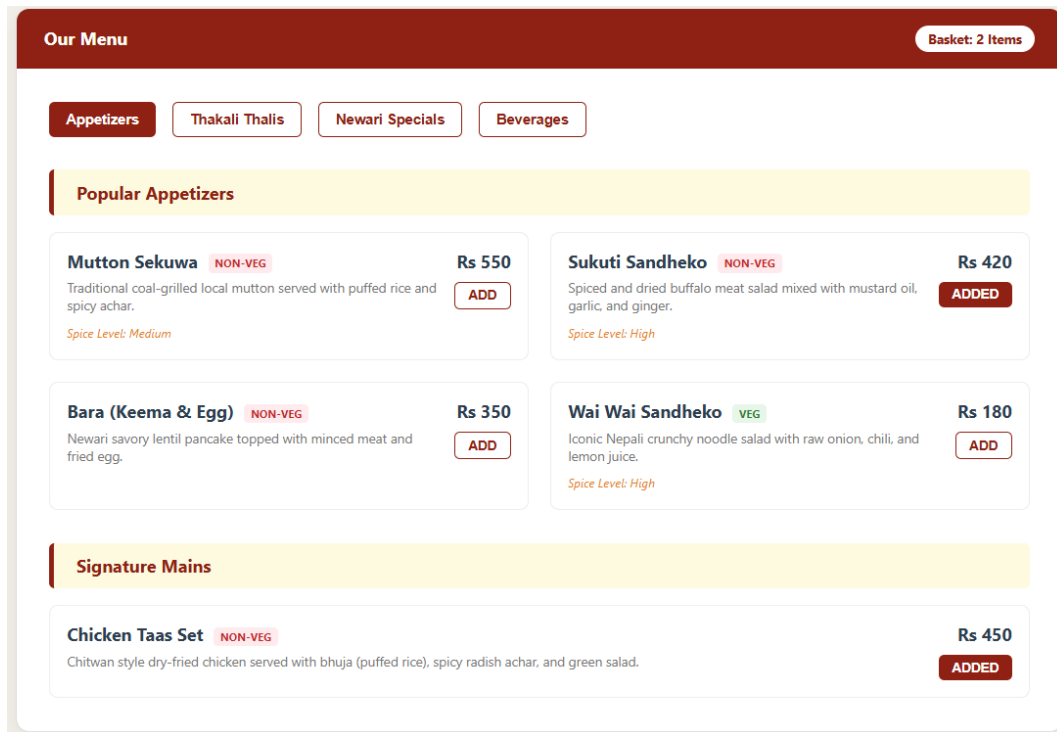
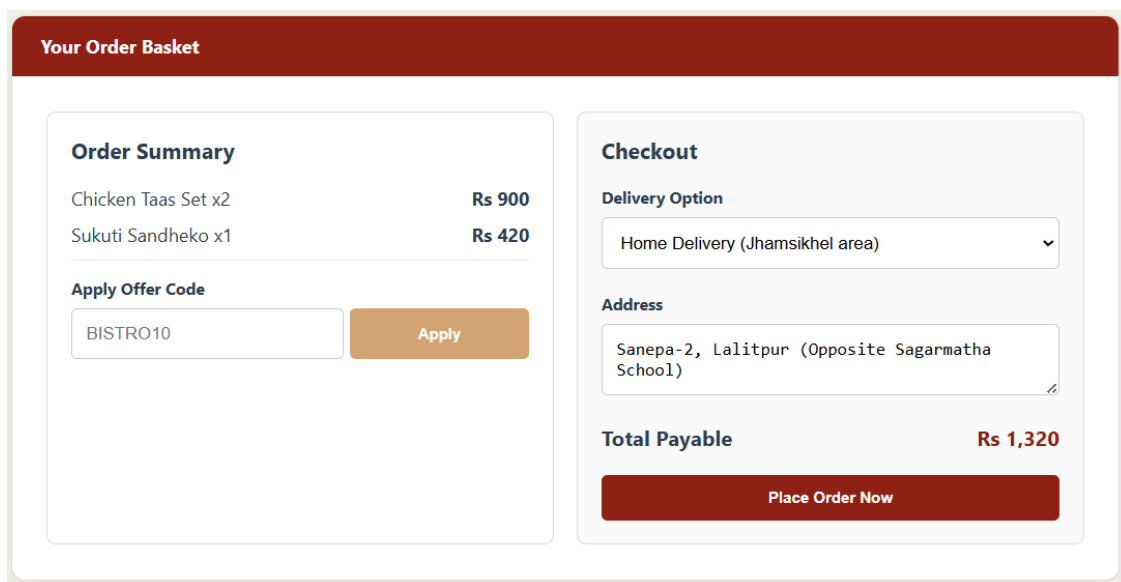


Figure 17: Menu browsing and item selection for members.

8.10 Prototype 10



Prototype 1: order basket, offer application, delivery option and checkout.

8.11 Prototype 11

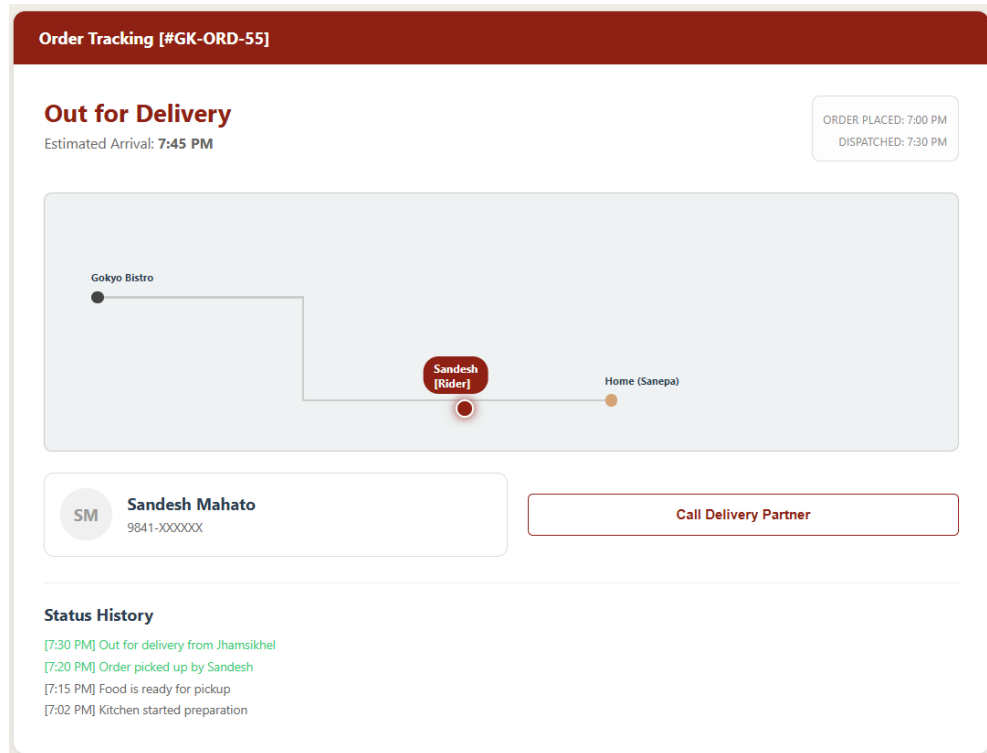


Figure 18: Home delivery tracking screen after order confirmation.

8.12 Prototype 12

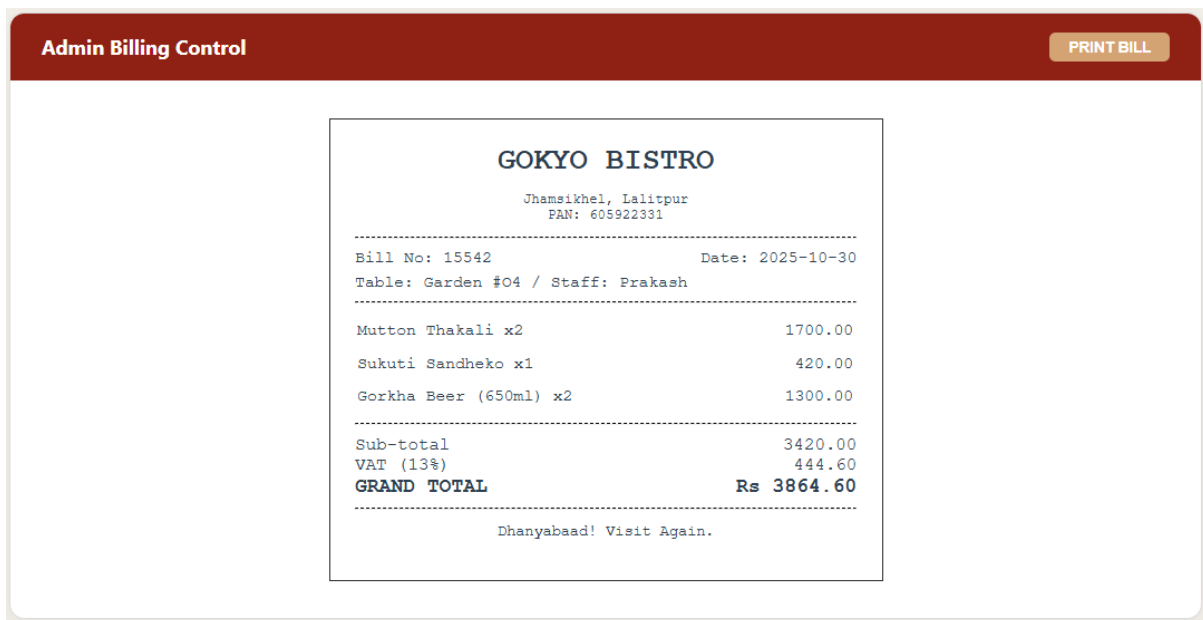


Figure 19: Bill and receipt generation screen with admin billing controls.

8.13 Prototype 13

Namaste, Rate Us!

How was your meal?

We value your feedback on your dining at Gokyo Bistro.

Food Quality (Taste & Portion)
Rating: 5 / 5

Service (Prakash & Team)
Rating: 4 / 5

Garden Ambience
Rating: 5 / 5

Tell us more...

The Mutton Sekuwa was amazing!

Submit Review

Figure 20: Rating and feedback screen for food, staff, ambience and delivery.

8.14 Prototype 14

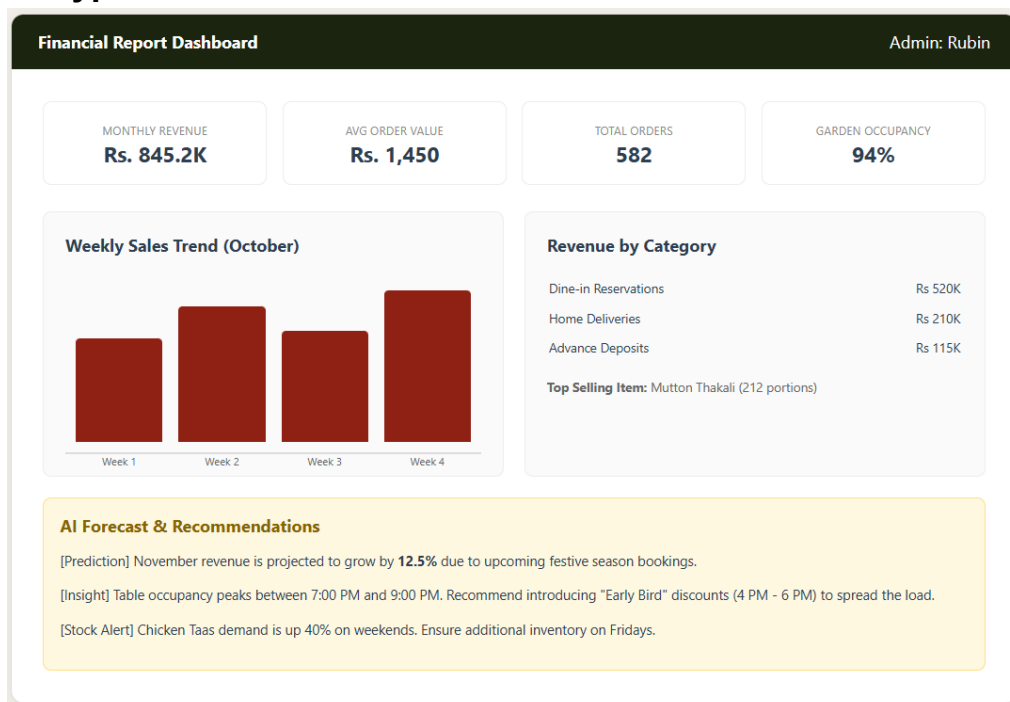


Figure 21: Admin financial report dashboard with revenue prediction and recommendations.

8.15 Prototype 15

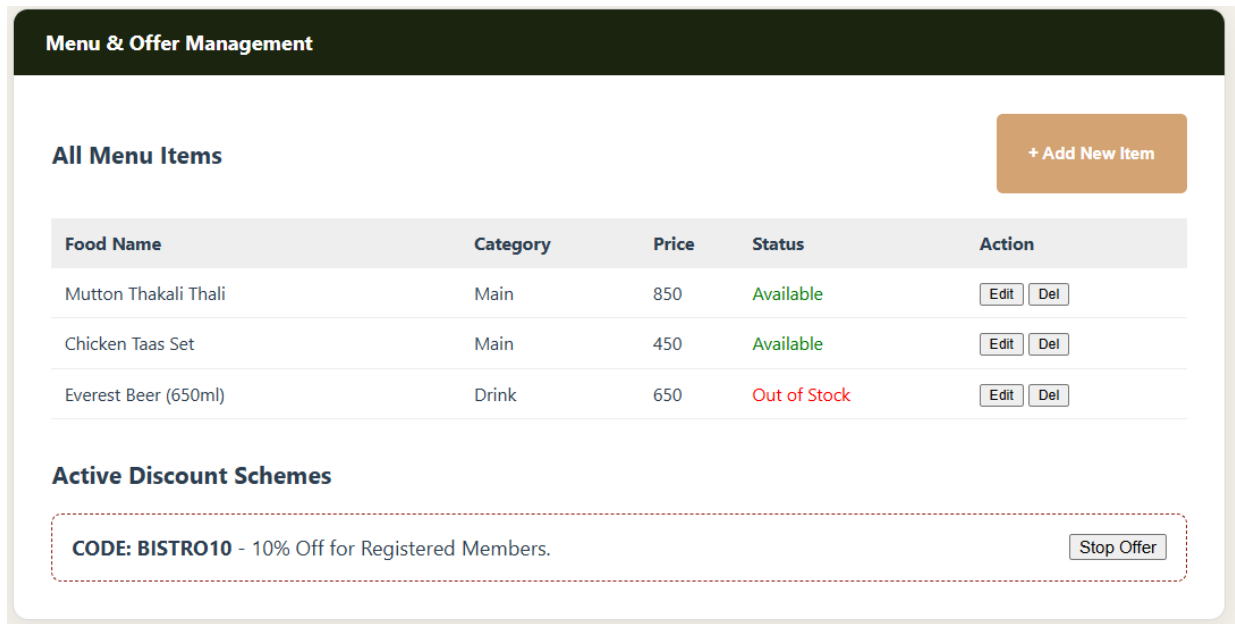


Figure 22: Admin menu CRUD, offers and discount scheme management.

9. Conclusion

The Gokyo Bistro Restaurant Management System is feasible as a layered MVC web application delivered using DSDM. The project should proceed from the completed analysis/design work into DSDM Foundations and Evolutionary Development. Priority should be given to must-have functions: registration/login, table availability, table booking, advance payment, cancellation/refund rules, menu browsing, order placement, home delivery, bill generation and admin menu control. Reporting, revenue prediction, recommendations and enhanced offers can then be refined in later timeboxes.

Testing should start at the beginning of development planning through acceptance criteria and test design. Black-box testing validates requirements and user flows, white-box testing validates internal service logic, and regression testing protects existing functionality after every change. Maintenance should be planned from deployment so that defects, security updates, database backups and future improvements can be handled in a controlled way.

10. References

Agile Business Consortium Chapter 6: Process. DSDM Project Framework. Available at:
<https://www.agilebusiness.org/dsdm-project-framework/process.html>

Agile Business Consortium 10: MoSCoW Prioritisation. DSDM Project Framework. Available at:
https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation

Agile Business Consortium Chapter 11: Iterative Development. DSDM Project Framework.
Available at: <https://www.agilebusiness.org/dsdm-project-framework/iterative-development.html>

Agile Business Consortium Chapter 12: Modelling. DSDM Project Framework. Available at:
<https://www.agilebusiness.org/dsdm-project-framework/modelling.html>

Agile Business Consortium (n.d.-e) What is DSDM? Discover the Eight Principles of DSDM.
Available at: <https://www.agilebusiness.org/business-agility/what-is-dsdm.html>

ISTQB (2024) Certified Tester Foundation Level (CTFL) v4.0 Overview. Available at:
<https://www.istqb.org/certifications/certified-tester-foundation-level-ctfl-v4-0/>