

# **Development Plan for the Ciel Consulting Online Ordering App: A Comprehensive Guide**

**(BA-Anand Vijaykumar Zanwar)**

## **1. Approach for the BA work:**

### **Stakeholder Identification and Requirement Gathering**

- Objective: Identify all key stakeholders such as users, business owners, and technical teams.
- Approach: Conduct interviews and workshops to gather the high-level needs and expectations for the online ordering app.

#### **1. Requirement Categorization**

**Objective:** Organize the requirements into major categories:

- Functional Requirements: Order placement, product browsing, payment options, and order tracking.
- Non-Functional Requirements: Cross-platform compatibility (laptops, tablets, mobile phones), global operating system support, and minimum 2G network requirement.
- Security Requirements: Secure user login, data encryption, and secure payment processing.
- Usability Requirements: Easy-to-use interface, accessible across different devices.

#### **2. Process Flow and Use Case Identification**

- Objective: Define the key workflows (e.g., placing an order, tracking orders) through use case diagrams and process flows.
- Approach: Map out the user journey from product selection to order confirmation and delivery.

### **3. Requirement Prioritization**

- Objective: Prioritize requirements based on criticality (e.g., Must-have features like order placement, Should-have features like multi-language support).
- Approach: Use MoSCoW prioritization to identify key features that need to be delivered first.

### **4. Collaborative Prototyping and Validation**

- Objective: Work with design and development teams to create early wireframes or prototypes.
- Approach: Validate these prototypes with stakeholders to ensure alignment with user expectations and business goals.

### **5. Testing and Feedback Collection**

- Objective: Ensure the app meets all the defined requirements through thorough testing (functional, security, usability).
- Approach: Facilitate User Acceptance Testing (UAT) to gather feedback and refine the app before launch.

## **2.Techniques for Business Analysis**

### **1. Process Modeling**

- **Description:** Creating visual representations of the end-to-end business processes (e.g., how a customer places an order, how it is processed, and how the order is fulfilled).
- **Purpose:** Helps identify inefficiencies, bottlenecks, and the complete flow of operations.

### **2. Use Cases**

- **Description:** Defining specific interactions between the users (actors) and the system (app) to achieve a particular goal (e.g., placing an order, checking order status).
- **Purpose:** Helps clarify system functionality from the user's perspective and ensures the system meets user needs.

### **3. Document Analysis**

- **Description:** Reviewing existing documentation, such as current ordering systems, company policies, or previous projects to gather insights on requirements.
- **Purpose:** Provides a historical context and helps identify gaps in the new system requirements.

### **4. Requirement Workshops**

- **Description:** Facilitating collaborative sessions with key stakeholders (users, business owners, technical teams) to discuss and gather detailed requirements.
- **Purpose:** Encourages open dialogue, identifies critical requirements, and ensures stakeholder alignment.

### **5. Interface Analysis**

- **Description:** Examining the interactions between the app and other systems, devices, or users (e.g., payment gateways, inventory systems).
- **Purpose:** Ensures smooth integration and seamless user experiences across different platforms.

### **6. Prototyping**

- **Description:** Creating early visual representations or models of the app interface to gather feedback from stakeholders before full development.
- **Purpose:** Helps identify design flaws early, reduces rework, and aligns expectations for the app's look and functionality.

### **7. SWOT Analysis (Strengths, Weaknesses, Opportunities, Threats)**

- **Description:** A strategic planning technique used to identify internal strengths and weaknesses of the app, as well as external opportunities and threats.
- **Purpose:** Helps ensure that the app is positioned for success in the competitive market and mitigates potential risks.

### **8. Brainstorming**

- **Description:** Generating a wide range of ideas from stakeholders and team members to explore all possible requirements, features, and solutions.

- **Purpose:** Encourages creative thinking and helps identify potential features or requirements that may not have been considered otherwise.

## 9. Focus Groups

- **Description:** Conducting sessions with a selected group of users or stakeholders to gather feedback and opinions on specific aspects of the app (e.g., usability, functionality).
- **Purpose:** Helps understand user expectations and preferences.

## 10. Benchmarking

- **Description:** Comparing the app's planned features with those of competitors or industry standards.
- **Purpose:** Ensures the app meets or exceeds market expectations and delivers competitive functionality.

## 3.Deliverables, Timeline, and Budget

### A) Deliverables

<b>Deliverable</b>	<b>Description</b>	<b>Due Date</b>
Project Plan	Detailed project plan outlining scope, key milestones, timeline, and stakeholders.	Week 1
Requirement Documentation	Comprehensive document containing functional, non-functional, technical, and security requirements.	Week 2
Process Flow Diagrams & Use Cases	Visual representations of user interactions and business processes.	Week 3
Wireframes/Prototypes	Initial user interface designs to demonstrate app functionality and design.	Week 4
Technical Specifications Document	Detailed technical requirements, including APIs,	Week 5

	integrations, and system architecture	
Security Plan	Outline of security measures such as authentication, encryption, and secure payment methods	Week 6
Testing Strategy and Test Cases	Plan for testing app functionality, security, and performance, including UAT	Week 7
Development Phase 1 (MVP)	MVP of the app with basic features like product browsing etc	Week 8-10
User Acceptance Testing (UAT)	Stakeholders and end users test the app for feedback and functionality validation.	Week 11
Final App Deployment	Full deployment of the app with all features	Week 12
Post-Launch Support Plan	Plan for ongoing maintenance, bug fixing, and feature updates post-launch.	Week 13

## B) Timeline

Phase	Duration	Key Activities
Planning and Requirement Gathering	Week 1- Week 2	Stakeholder meetings, requirement workshops, process modeling, and document analysis.
Design and Prototyping	Week 3 – Week 4	Develop use cases, wireframes, and UI prototypes based on requirements
Technical Specification & Security Planning	Week 5- Week 6	Create technical documentation, API specifications, and security measures
Development (MVP)	Week 6 – Week 7	App development (core features: product browsing, order placement, payments

Testing & User Acceptance Testing	Week 8 – Week 9	Functional, security, and performance testing; UAT with stakeholders.
Final App Deployment	Week 10- Week 11	Full release of the app on all platforms
Post-Launch Support	Week 13	Bug fixing, user feedback incorporation, and feature updates.

### C) Estimated Budget

Category	Estimated Cost	Description
Requirement Gathering & Analysis	\$ 5,000	BA work, stakeholder meetings, requirement documentation.
Design & Prototyping	\$ 8,000	UI/UX design, wireframing, and prototyping for multi-device interfaces
Development (MVP)	\$ 25,000	App development (core functionality for all devices and operating systems).
Security Implementation	\$ 7,000	Data encryption, user authentication, secure payment gateways, and testing.
Testing (QA & UAT)	\$ 10,000	Testing the app across all platforms and conducting user acceptance testing.
Final Deployment & Post-Launch Support	\$ 5,000	App launch, initial bug fixes, and ongoing support post-launch.
Contingency (10%)	\$ 6,000	Buffer for unforeseen costs and additional features

**Total Estimated Cost                  = \$ 66,000**

## 4. Formality/the level of details

### 1. Formal Requirement Documentation

- **Details:** A comprehensive Business Requirements Document (BRD) will be created, capturing all functional and non-functional requirements. Each requirement will be traced back to its source (stakeholder, market analysis) and assigned a unique identifier for easy reference during development and testing.
  
- **Contents:**
  - Introduction and project overview
  - Objective and business goals
  - Detailed functional and non-functional requirements
  - Security, usability, and performance requirements
  
  - Compliance and regulatory standards
- **Format:** A formalized structure with approval signatures from all stakeholders to ensure consensus.

### 2. Change Request Process

- **Details:** Establish a Change Request Form and a strict process to handle any modifications to requirements or scope. Each change will undergo a formal review for its impact on budget, timeline, and resources.
- **Approval Flow:** Stakeholder identification > Impact analysis > Approval/Reject > Implementation.

### 2. Project Plan and Timeline Documentation

#### 1. Formal Project Plan

- **Details:** A formal project management plan will be developed, outlining key milestones, dependencies, risks, and timelines. Each milestone will be broken down into tasks with assigned roles and responsibilities.
- **Contents:**
  - Scope definition
  - Resource allocation and team structure

- Timeline with Gantt chart or other visual aids for tracking progress
  - Risk management and contingency planning
  - Approval checkpoints
- **Format:** A detailed project plan document shared with all stakeholders via the project management tool (e.g., Jira, MS Project), with weekly review meetings scheduled.

### 3. Testing and Quality Assurance (QA)

#### 1. Test Plan and Test Cases

- **Details:** A formal test plan will be created, covering the testing strategy (unit testing, integration testing, UAT) and key test cases for each functional and non-functional requirement. This plan will be reviewed and signed off by key stakeholders before testing begins.
- **Contents:**
  - Types of tests to be conducted (functional, security, performance)
  - Test environment and tools
  - Entry and exit criteria
  - Test cases, success/failure criteria, and defect tracking process

### 4. Security and Compliance Formality

#### 1. Security Plan

- **Details:** A formal security plan will be developed that includes:
  - Secure login mechanisms (2-factor authentication)
  - Encryption standards for data in transit and at rest
  - Compliance with GDPR, PCI-DSS, and other global data protection laws.
- **Approvals:** Security experts will formally sign off on the app's security measures, with regular audits to ensure compliance throughout the development lifecycle.

## 5. Formal Stakeholder Reporting

### 1. Progress Reporting

- **Details:** Regular formal status reports will be submitted to stakeholders, including weekly and monthly progress updates, budget tracking, and risk assessments.
- **Format:** These reports will follow a formal template, ensuring consistency and clarity, including sections for:
  - Completed tasks and upcoming milestones
  - Budget vs actual spend
  - Identified risks and mitigation strategies
  - Issues and resolutions

## 6. Formal Sign-Off and Acceptance Criteria

### 1. Deliverable Sign-Off

- **Details:** Each deliverable (requirement document, design, MVP, final app) will require formal sign-off from the relevant stakeholders. No deliverable will be moved to the next phase without approval.
- **Acceptance Criteria:** Formal acceptance criteria will be documented and agreed upon during the requirement-gathering phase, ensuring all parties have a clear understanding of what constitutes successful completion.

## 7. Detailed Budget Tracking

### 1. Budget Allocation & Tracking

- **Details:** A formal budget will be allocated and tracked for each phase of the project. Monthly or bi-monthly budget reports will be shared with stakeholders, providing an up-to-date financial snapshot.
- **Format:** Budget breakdown will include:
  - Resource costs (developers, testers, designers)
  - Infrastructure costs (servers, third-party services)
  - Contingency funds for unexpected costs

## **8. Formal Training and Post-Launch Support**

### **1. User Training Plan**

- **Details:** A formal training plan will be developed, including user guides, video tutorials, and live sessions.
- **Post-Launch Support:** A post-launch support team will be formally established to address any bugs or issues identified after the app goes live, with a Service Level Agreement (SLA) in place for resolving critical issues.

## **5. Requirement prioritization approach:**

The MoSCoW Prioritization Technique will be applied to classify requirements into four categories:

- **Must-Have:** Critical and non-negotiable features that the app cannot function without.
- **Should-Have:** Important but not essential features that add significant value and can be included if possible.
- **Could-Have:** Desirable features that can enhance the user experience but are not critical for the app's core functionality.
- **Won't-Have (This Time):** Features that are out of scope for this phase and can be deferred to future versions.

## **2. Prioritization Criteria**

Requirements will be prioritized based on several factors, including:

- **Business Impact:** The potential of the requirement to fulfill the business objectives and drive revenue or efficiency.
- **User Needs:** The level of importance from a user's perspective, enhancing customer satisfaction and usability.
- **Technical Complexity:** The difficulty or simplicity in implementing the requirement, considering available resources.
- **Time-to-Market:** The urgency of certain features to meet deadlines, such as for a product launch or competitive advantage.

- **Cost:** The financial impact of implementing a particular feature and its return on investment.
- **Regulatory and Security Compliance:** Features necessary to meet legal, regulatory, or security requirements.

### 3. Stakeholders Involved in Prioritization

#### **Key Stakeholders:**

##### **1. Project Sponsor**

- **Role:** Provides funding for the project and sets the high-level objectives. Ensures that the prioritization aligns with the company's business goals and financial resources.

##### **2. Product Owner**

- **Role:** Acts as the primary decision-maker for product features and prioritization. Responsible for ensuring that the app delivers maximum value to the business and users.

##### **3. Development Team (Tech Leads)**

- **Role:** Provides insights into the technical feasibility and complexity of requirements. Helps identify dependencies and potential technical risks that could affect prioritization.

##### **4. Business Analyst (BA)**

- **Role:** Facilitates discussions, analyzes business needs, and works closely with stakeholders to ensure the requirements are clear, well-defined, and properly prioritized based on the business and user goals.

##### **5. Customer Service Team/End Users**

- **Role:** Offers feedback on user needs and pain points. Their perspective helps prioritize features that will most improve the user experience and customer satisfaction.

## **6. Marketing Team**

- **Role:** Informs which features are critical to the app's launch strategy and market positioning. Helps prioritize requirements that will drive user engagement and differentiation.

## **7. Security/Compliance Officers**

- **Role:** Ensures that all security and regulatory requirements are met. They identify features related to user data protection, payment security, and compliance with laws such as GDPR.

## **8. Finance Department**

- **Role:** Monitors costs and ensures that the prioritized features align with the available budget. Ensures that there are no cost overruns and features provide good ROI.

# **4. Requirement Prioritization Process**

## **1. Initial Requirement Gathering**

- The BA will first gather all requirements through techniques like workshops, use cases, and document analysis. These requirements will be listed and classified based on functional and non-functional categories (e.g., security, performance, usability, etc.).

## **2. Scoring the Requirements**

- Each requirement will be scored based on the prioritization criteria: Business Impact, User Needs, Technical Complexity, Time-to-Market, Cost, and Compliance.
- Scores will be assigned on a scale (e.g., 1-5), with higher scores indicating higher importance.

## **3. MoSCoW Classification**

- Based on the scores and discussions with stakeholders, each requirement will be placed into the MoSCoW categories (Must, Should, Could, Won't have).

#### **4. Stakeholder Review Meetings**

- Regular prioritization meetings will be held with key stakeholders (Product Owner, Tech Leads, BA, etc.) to discuss the scores, adjust priorities as necessary, and finalize the order of implementation.

#### **5. Final Prioritization and Sign-Off**

- Once the stakeholders reach a consensus, the final list of prioritized requirements will be signed off by the Product Owner and Project Sponsor.

#### **6. Ongoing Review and Adjustments**

- The prioritized list will be revisited at regular intervals (e.g., at the end of each sprint or phase) to accommodate changes based on new information, market conditions, or technical challenges.

### **5. Example of Prioritized Requirements**

Requirement	MoSCoW Category	Stakeholders Involved
Secure user authentication (e.g., 2FA)	Must-Have	Security Officers, BA
Payment gateway integration	Must-Have	Product Owner, Finance
Cross-device compatibility (mobile, tablet, PC)	Must-Have	Development Team, Product Owner
Order tracking feature	Should-Have	Customer Service, End Users
Push notifications for order updates	Could-Have	Marketing, End Users
Social media login integration	Could-Have	Marketing, Development Team
Advanced product recommendations engine	Won't-Have	Product Owner, Marketing

## 6. Tools for the BA work:

### 1. Jira

- **Purpose:** Jira will be used as the primary tool for tracking requirements, user stories, tasks, and issues. It allows for clear visibility into the progress of each requirement and facilitates communication between stakeholders.
- **Features:**
  - Agile boards for sprint management
  - Custom workflows for different stages of requirements (e.g., "To Do," "In Progress," "Done")
  - Linking requirements to development tasks, bugs, and test cases
  - Real-time tracking and reporting for stakeholders

### 2. Confluence

- **Purpose:** Confluence will serve as a central repository for documenting all business requirements, user stories, meeting notes, and decision logs. It ensures that all stakeholders have access to up-to-date information throughout the project lifecycle.
- **Features:**
  - Collaborative editing for team members
  - Spaces to organize project documents by category (e.g., Requirements, Designs, Meetings)
  - Version control to track changes made to requirements and documents
  - Integration with Jira for seamless issue tracking and linking of requirements

### 3. Microsoft Excel/Google Sheets

- **Purpose:** Excel or Google Sheets will be used to create and maintain a Requirement Traceability Matrix (RTM) to ensure that all requirements are fulfilled throughout the development process.

- **Features:**
  - Customizable templates to capture requirements, their priority, and their status
  - Filtering and sorting options to track specific requirements
  - Easy-to-use for quick updates and sharing with stakeholders

## 2. Collaboration and Communication Tools

### 1. Slack

- **Purpose:** Slack will be used for real-time communication among team members, enabling quick discussions and resolution of questions related to requirements and development.
- **Features:**
  - Dedicated channels for specific aspects of the project (e.g., "#requirements," "#development," "#testing")
  - Direct messaging and video calls for quick, informal discussions
  - Integration with Jira for real-time notifications about changes or updates

### 2. Microsoft Teams

- **Purpose:** For formal meetings and document sharing, Microsoft Teams will be used. It provides video conferencing, file sharing, and chat features, allowing for structured communication and collaboration across teams.
- **Features:**
  - Video and audio meetings for requirement review sessions
  - Integration with Microsoft Office tools like Word and Excel
  - Secure document sharing and collaboration
  - Recording meetings for future reference

## 3. Documentation and Reporting Tools

### 1. Microsoft Word/Google Docs

- **Purpose:** Detailed requirement documents, Business Requirements Documents (BRD), and other formal reports will be created using Microsoft Word or Google

Docs. These tools are suitable for formalized, structured documents that need to be distributed to stakeholders.

- **Features:**
  - Customizable templates for BRDs, project charters, etc.
  - Real-time collaboration in Google Docs
  - Formatting features for formal documentation
  - Commenting for review and feedback from stakeholders

## 2. Power BI/Tableau

- **Purpose:** Reporting and data visualization tools like Power BI or Tableau can be used to create dashboards that track the project's progress, budget, and requirement fulfillment.
- **Features:**
  - Interactive dashboards to monitor key performance indicators (KPIs)
  - Data connections to Jira or other tools for real-time reporting
  - Visual insights into project status, such as completion percentage of requirements or budget usage

## 4. Wireframing and Prototyping Tools

### 1. Balsamiq

- **Purpose:** Balsamiq will be used to create low-fidelity wireframes that represent the visual structure of the online ordering app, helping to clarify user interface (UI) and user experience (UX) requirements.
- **Features:**
  - Quick and easy wireframe creation for early-stage prototyping
  - Collaborative feedback from stakeholders on the design layout
  - Supports linking wireframes with requirements in Jira or Confluence

### 2. Figma

- **Purpose:** Figma will be used for high-fidelity prototyping and interactive designs, especially when demonstrating complex workflows or interactions within the app.

- **Features:**
  - Collaborative design environment for stakeholders to provide real-time feedback
  - Interactive prototypes to simulate user journeys and actions
  - Integration with Confluence for sharing design documents

## 5. Testing and Quality Assurance Tools

### 1. TestRail

- **Purpose:** TestRail will be used to manage test cases, track testing progress, and ensure that all requirements are thoroughly tested during User Acceptance Testing (UAT) and other test phases.
- **Features:**
  - Organizes test cases by requirement or feature
  - Tracks test progress and defect management
  - Integration with Jira for linking test results with issues

### 2. Selenium (Automated Testing)

- **Purpose:** Selenium will be used for automated testing of the app's UI and core functionalities to ensure that the requirements are met without manual intervention.
- **Features:**
  - Automated test scripts for regression testing
  - Cross-browser testing to ensure compatibility across different devices
  - Reporting on test pass/fail outcomes

## 7. Project complexity:

### 1. Number of Impacted Areas:

This project impacts multiple areas within Ciel Consulting's operations:

- **Sales and Order Management:** The app will directly impact how orders are placed, processed, and managed within the company's sales pipeline, requiring changes to existing processes.
- **Inventory Management:** The online ordering app will need to interface with the inventory system to ensure real-time stock availability for customers placing orders
- **Customer Relationship Management (CRM):** Integration with CRM systems will be essential for managing customer interactions, tracking order history, and providing personalized services like order status notifications.
- **Payment Systems:** The app will involve integrating secure payment gateways, affecting financial operations and requiring robust security measures.
- **IT Infrastructure and Support:** Supporting the app across multiple devices, operating systems, and network conditions (2G and higher) will require updates to Ciel Consulting's IT infrastructure, including servers, cloud solutions, and help desk operations.
- **Marketing and User Engagement:** The app will enable personalized marketing and push notifications, requiring changes to the marketing strategy and systems for customer engagement.

### 2. Criticality of the Change from an Organizational Perspective

This project is highly critical for Ciel Consulting for several reasons:

- **Strategic Importance:** The online ordering app is crucial for expanding Ciel Consulting's digital presence and reaching customers globally. Given the increasing shift to online purchases, this app will become a primary revenue channel.
- **Competitive Advantage:** Launching this app across all devices and operating systems, with the ability to work even on 2G networks, offers a significant

competitive advantage, allowing the company to cater to a broader market, especially in areas with limited internet access.

- **Customer Satisfaction and Experience:** A seamless, cross-platform online ordering experience is central to enhancing customer satisfaction. Failure to deliver a smooth user experience could lead to a loss of customers and damage the company's reputation.
- **Operational Efficiency:** The app will streamline internal processes like order collection, tracking, and payments, improving overall efficiency. Any failure in these processes could disrupt daily operations and result in financial loss.
- **Security and Compliance:** Given the handling of sensitive customer information and payments, the project involves strict security and regulatory compliance, making it a high-stakes change for the company.

### 3. Technical and Implementation Complexity

Several technical challenges contribute to the complexity of the project:

- **Cross-Device and Cross-Platform Compatibility**  
The app must function seamlessly on various devices (laptops, tablets, smartphones) and operating systems (iOS, Android, Windows, etc.). Ensuring consistent performance across these platforms adds complexity.
- **Network Constraints:** Designing an app that can run smoothly even on a 2G network introduces additional challenges in terms of optimizing performance, speed, and data usage.
- **Integration with Existing Systems:** The app needs to integrate with multiple existing systems like inventory management, CRM, and payment gateways, adding complexity in terms of ensuring real-time data sync and seamless workflows.
- **Security Requirements:** Secure handling of user authentication, data encryption, and payment processing will require advanced security protocols and compliance with standards like PCI DSS and GDPR, adding to the technical complexity.

#### **4. Stakeholder Involvement**

The number of stakeholders involved increases the project's complexity:

- **Internal Stakeholders:**
  - Project Sponsor
  - Sales Team
  - Marketing Team
  - IT and Infrastructure Teams
  - Customer Service
- **External Stakeholders:**
  - App Developers
  - Payment Gateway Providers
  - Third-party Vendors (e.g., hosting, cloud services)
  - Regulatory Bodies

Managing expectations and gathering requirements from this broad range of stakeholders will require significant coordination and communication, adding complexity to the project.

#### **Conclusion: Overall Complexity Assessment**

Based on the factors discussed, the overall project complexity is high due to:

- The large number of impacted business areas, including sales, CRM, inventory, payments, and IT.
- The critical nature of the app for business growth, customer satisfaction, and operational efficiency.
- The technical complexity related to cross-platform compatibility, network constraints, and security requirements.
- The involvement of both internal and external stakeholders, each with their own requirements and expectations.

## 8. Approach to scope and change management

### Define and Document Scope Early

- **Initial Scope Definition:** The initial project scope will be defined based on the high-level business requirements, including:
  - User ability to place orders using any internet-enabled device
  - Compatibility with all major operating systems
  - Functionality on 2G and higher network speeds
- **Scope Baseline:** A scope baseline will be established, consisting of the project's Business Requirements Document (BRD), Functional Specifications, and User Stories. This will act as a point of reference for all future change requests.

### Requirements Traceability

- A Requirements Traceability Matrix (RTM) will be used to track each requirement, ensuring that every change is linked to a business need and properly implemented during development.
- Jira will be used to document and track changes in scope and ensure that the impact on the timeline and resources is clear.

### Regular Scope Reviews

- **Scope Review Meetings:** Regular scope review meetings with stakeholders (e.g., business teams, IT, and developers) will be scheduled to ensure alignment on the project's current scope and any changes that arise.
- **Scope Control:** Any changes to the scope will be evaluated based on their impact on project timelines, resources, and overall objectives. Only changes that are critical or add significant business value will be considered.

---

## 2. Change Management Approach

### Change Request Process

- **Formal Change Requests:** Any changes to requirements, functionality, or design will be managed through a formal Change Request (CR) process. Stakeholders must submit a CR to the project manager or business analyst for evaluation.

### Impact Assessment

- **Change Impact Analysis:** Each change request will go through an impact analysis to assess its:
  - Time Impact: How will this change affect the project timeline?
  - Resource Impact: Will additional resources (e.g., development or testing teams) be required?
  - Cost Impact: Will this change increase the budget? If so, by how much?
  - Scope Impact: Does this change require additional features or processes outside of the original project scope?

### Change Control Board (CCB)

- A Change Control Board (CCB) will be established, consisting of key stakeholders, such as:
  - Project Sponsor
  - Business Analyst
  - Project Manager
  - IT Lead
  - Marketing/Customer Experience Lead
- The CCB will be responsible for reviewing and approving or rejecting change requests. If approved, the change will be integrated into the project plan and communicated to all stakeholders.

### Communication and Documentation of Changes

- **Change Log:** All approved changes will be recorded in a change log that will be maintained in Jira and shared with relevant stakeholders. The log will include details of the change, the rationale for the change, and the expected impact.
- **Update Documentation:** All project documentation, including requirements, project plans, and timelines, will be updated to reflect the changes.
- **Stakeholder Notification:** Any major changes will be communicated through email and project meetings to ensure that all stakeholders are aligned with the updated scope.

## 3. Use of Flowcharts or High-Level Process Diagrams

To visualize and communicate the process of managing scope and changes effectively, a high-level flowchart will be used. This will show how change requests move through the system from submission to approval or rejection. Below is an outline of the flowchart:

Change Request Process Flowchart (High-Level)

1. **Change Request Submitted:** Stakeholders submit a formal CR.

2. **Initial Evaluation by BA/PM:** The Business Analyst and Project Manager assess the request for clarity and initial feasibility.
3. **Impact Analysis:** The team conducts an impact analysis covering time, cost, resources, and scope.
4. **Change Control Board Review:** The CCB reviews the change request and the impact analysis.
  - If Approved: The change is accepted and documented, and the project plan is updated.
  - If Rejected: The request is closed, and feedback is provided to the requester.
5. **Implementation:** The approved change is implemented by the development team, and relevant documentation is updated.
6. **Testing and Validation:** The change is tested and validated to ensure it meets the updated requirements
7. **Communication to Stakeholders:** The team communicates the change implementation to all stakeholders.

#### **4. Handling Scope Creep**

To avoid scope creep (uncontrolled expansion of the project scope without corresponding time, cost, or resource adjustments):

- **Strict Adherence to Change Control Process:** All changes, no matter how small, must go through the formal change request process.
- **Early Identification of "Nice-to-Have" Features:** During requirement gathering, "nice-to-have" features that do not impact the core functionality of the app will be identified and postponed to a future phase if necessary.
- **Regular Stakeholder Engagement:** Regular meetings with stakeholders will help manage expectations and ensure that additional features or changes are evaluated properly before being included in the project.

## 9. Approach to sign-off

### 1. Define Clear and Detailed Requirements

- **Requirements Specification Document (RSD):** All functional and non-functional requirements will be clearly documented in a Requirements Specification Document (RSD). This document will include:
  - **Business Requirements:** High-level goals the app should achieve.
  - **Functional Requirements:** Detailed features like user interface, device compatibility, order processing, and security
  - **Non-Functional Requirements:** Performance, scalability, and network capabilities (including 2G).
  - **Acceptance Criteria:** Define specific, measurable conditions that must be met for each requirement to be considered complete.
- **Early Stakeholder Involvement:** Key stakeholders, including product owners, IT teams, and business sponsors, will be engaged early to review drafts of the requirements to address concerns before formal approval.

### 2. Review and Validation Process

- **Requirements Review Meetings:** Organize multiple rounds of review sessions with different stakeholder groups (technical, business, operations) to gather feedback, clarify doubts, and ensure understanding of the documented requirements
- **Collaborative Tools:** Use tools like Jira and Confluence to provide a central platform for stakeholders to track changes, ask questions, and suggest modifications to requirements in real time.
- **Feedback Incorporation:** Any necessary revisions or adjustments based on stakeholder feedback will be made promptly to ensure all concerns are addressed.

### **3. Sign-Off Workflow**

- **Final Review and Walkthrough:** After all feedback is incorporated, a final walkthrough will be held where the Business Analyst will explain each requirement, ensuring stakeholders fully understand the scope and objectives.
- **Formal Approval via Sign-Off Document:**
  - A formal Sign-Off Document will be created and distributed to all key stakeholders, summarizing the final requirements and outlining the scope, timeline, and responsibilities.
  - Stakeholders will be required to provide their formal approval via digital signatures (using tools like DocuSign) or email confirmation.
- **Stakeholders Involved in Sign-Off:**
  - Project Sponsor
  - Business Owners
  - IT/Development Leads
  - Product Owner
  - QA/Test Teams

### **4. Change Control Post Sign-Off**

- **Formal Change Requests:** After sign-off, any changes to the requirements must go through the formal change request process. This ensures that modifications are carefully evaluated for their impact on timelines, budget, and scope.
- **Updated Documentation:** If a change is approved, the RSD will be updated, and stakeholders will sign off again on the revised version to ensure continued alignment.

### **5. Communication and Documentation**

- **Distribution of Final Signed Document:** Once sign-off is complete, the final RSD and sign-off record will be distributed to all stakeholders and stored in a central repository (e.g., Confluence or SharePoint) for future reference.
- **Regular Status Updates:** Regular project status meetings will be held to ensure that the development team is working according to the signed-off requirements and that stakeholders are aware of any potential issues or risks.

## **10. Approach to communication**

### **1. Communication Mediums**

The following communication channels will be used to ensure seamless collaboration between all stakeholders:

- **Email:** For formal updates, documentation sharing, meeting invites, and follow-ups. All significant project updates will be communicated via email to ensure there is a record of important decisions and actions.
- **Instant Messaging (Slack/Microsoft Teams):** For quick, real-time communication between team members, developers, and the project management team. This will be used for informal discussions and quick resolutions to queries.
- **Video/Conference Calls (Zoom/Microsoft Teams):** For regular meetings, including requirements discussions, project status updates, and sprint reviews. This will be used for high-level discussions, walkthroughs, and to address any concerns in real time.
- **Project Management Tools (Jira/Confluence):** For tracking project progress, task assignments, documenting changes, and collaborating on project requirements. Jira will be used for tracking tasks, bugs, and any issues that arise, while Confluence will serve as a repository for shared documents and project notes.

### **2. Communication Frequency**

The frequency of communication is based on the project's phase and the importance of keeping stakeholders updated. Here's a breakdown:

#### **Regular Updates**

- **Weekly Status Meetings:**
  - **Frequency:** Once a week.
  - **Attendees:** Project manager, business analyst, developers, and key stakeholders.

- **Purpose:** To review project progress, discuss any issues or risks, and track milestones. This will keep all teams updated on the current status of the project and ensure alignment.
- **Daily Stand-Ups (Agile):**
  - **Frequency:** Daily (15-30 minutes).
  - **Attendees:** Development team, business analyst, project manager.
  - **Purpose:** These short stand-up meetings will focus on daily task progress, blockers, and next steps to ensure the team is on track.
- **Bi-Weekly Stakeholder Reports:**
  - **Frequency:** Every two weeks.
  - **Medium:** Email and posted on project management tools (Jira/Confluence).
  - **Attendees:** All stakeholders, including the business team, IT team, and executive sponsors.
  - **Purpose:** To provide a summary of completed work, upcoming milestones, any changes to scope or timelines, and highlight critical decisions made.

## Milestone-Based Communication

- **Key Milestone Reviews:**
  - **Frequency:** At the completion of significant project phases (e.g., requirement gathering, design, development, testing, etc.).
  - **Purpose:** To review completed work for each phase, ensuring that it meets the required specifications before moving to the next phase.
  - **Medium:** Video meetings followed by detailed documentation shared via email and stored in project tools.
- **Ad-hoc Meetings:**
  - **Frequency:** As needed.
  - **Purpose:** To address urgent issues, review critical changes, or resolve challenges in real-time. These can be scheduled on short notice if needed.
  - **Medium:** Video calls or phone calls for urgent matters.

## 3. Key Stakeholders for Communication

- **Internal Project Team:** Development team, QA, business analysts, project managers, and designers.
- **External Stakeholders:** Business owners, product owners, marketing teams, and executive sponsors.
- **Customers/User Representatives:** Involvement in feedback cycles and user testing phases.

#### 4. Communication Types

- **Progress Updates:** Weekly updates on the project's current progress, timelines, and task completion.
- **Issue Escalation:** Urgent communication (via email or call) to inform key stakeholders of critical blockers or issues that may affect project delivery.
- **Requirement Clarification:** Immediate communication (via instant messaging or calls) to clarify any ambiguities or changes in the project requirements.
- **Change Requests:** Formal email or project management tool documentation will be used to request, assess, and approve changes to scope, timelines, or deliverables.
- **Risk Management:** Any identified risks or issues will be communicated via status reports, and in critical situations, escalated via email or calls to ensure timely resolution.

#### 5. Documentation

- **Project Artifacts:** All documents related to the project (such as requirement specifications, change logs, meeting minutes, etc.) will be stored in a central repository like **Confluence** for easy access and version control.
- **Meeting Minutes:** Every meeting, whether formal or informal, will have its minutes documented and shared with stakeholders for record-keeping and reference.

#### 6. Feedback and Collaboration

- **Feedback Loops:** Regular feedback will be gathered from key stakeholders and users, particularly during user testing phases, to ensure the app meets user needs. Tools like Google Forms or surveys will be used to gather structured feedback.
- **Collaboration:** Cross-functional teams will use shared platforms like Jira, Confluence, and Slack to collaborate efficiently, ensuring that communication is fluid and centralized

## 11. Stakeholders

### 1. Project Sponsor

- **Role:** The project sponsor is responsible for providing overall direction and ensuring that the project aligns with Ciel Consulting's strategic goals. They authorize the project and secure funding.
- **Involvement:** Key decision-maker for project scope, budget, and timeline. They will approve major milestones and provide final sign-off on the project deliverables.

### 2. Product Owner

- **Role:** Represents the business side, often responsible for defining the product vision, prioritizing features, and ensuring the development team builds the app in alignment with business goals.
- **Involvement:** Provides continuous feedback on the app's development, approves features, and makes decisions on product changes during the development cycle.

### 3. Development Team

- **Role:** Includes software engineers, front-end and back-end developers, and system architects responsible for building and deploying the app.
- **Involvement:** Responsible for coding, testing, and ensuring the app meets the technical requirements such as compatibility with multiple devices (laptops, tablets, smartphones) and networks (2G+).

### 4. Business Analyst (BA)

- **Role:** The Business Analyst gathers and documents the functional and non-functional requirements for the app, acts as the liaison between business and technical teams, and ensures that the project meets business needs.

- **Involvement:** Facilitates requirement workshops, models business processes, conducts gap analysis, and communicates requirements to the development team.

## 5. Quality Assurance (QA) Team

- **Role:** Responsible for testing the app to ensure it meets the required quality standards, including functionality, usability, performance, security, and compatibility across various devices and network conditions.
- **Involvement:** Designs and executes test cases, identifies defects, and ensures the app functions as expected before deployment.

## 6. UX/UI Designers

- **Role:** UX (User Experience) and UI (User Interface) designers are responsible for creating a seamless and user-friendly design for the app. This includes designing the user interface for laptops, tablets, and mobile phones.
- **Involvement:** Work with the BA and product owner to ensure the design meets user expectations and business goals. They also create wireframes and prototypes for the app's design.

## 7. Marketing Team

- **Role:** Responsible for developing strategies to promote the app and attract users.
- **Involvement:** Ensures the app aligns with Ciel Consulting's brand and oversees the go-to-market strategy, working with the development team to ensure features are designed with user engagement in mind.

## 8. Legal and Compliance Team

- **Role:** Ensures the app complies with all relevant legal and regulatory requirements, such as data privacy, consumer rights, and intellectual property laws.
- **Involvement:** Reviews the app for legal risks and ensures user agreements, data protection policies, and terms of service are clearly defined and implemented.

## 9. IT Infrastructure Team

- **Role:** Responsible for the underlying infrastructure and deployment of the app, including servers, cloud services, and network configurations.

- **Involvement:** Ensures the app can handle traffic, is secure, and has adequate server resources to operate efficiently under all expected network conditions (2G+).

## 10. Customer Support Team

- **Role:** Responsible for handling customer inquiries, troubleshooting issues, and providing feedback on user-reported problems.
- **Involvement:** Provides input on user workflows and customer needs, and helps design support mechanisms such as FAQs, chatbots, or customer service channels within the app.

## 11. End Users (Customers)

- **Role:** The ultimate users of the online ordering app who will place orders using various devices.
- **Involvement:** Provide feedback during user acceptance testing (UAT) and post-launch to ensure the app meets their needs in terms of usability, reliability, and performance.

## 12. Finance Team

- **Role:** Manages the project budget and financial aspects, ensuring the project stays within the approved financial constraints.
- **Involvement:** Monitors the financial impact of the project and may analyze cost projections for app development, maintenance, and future upgrades.

## 13. Operations/Logistics Team

- **Role:** Manages order fulfillment, shipping, and delivery processes that will be integrated with the app.
- **Involvement:** Provides input on operational workflows, ensuring that the app's ordering and fulfillment systems align with the logistics team's processes for efficient order processing.

## 14. External Vendors (If Applicable)

- **Role:** Third-party service providers for payment processing, network services, cloud hosting, or other functionalities that the app might need.
- **Involvement:** Collaborate with the development and IT teams to integrate their services seamlessly with the app.

## 12. Governance Approach for Approvals

Criteria	Project Objective	Success Criteria	Person Approving
<b>Scope</b>	Deliver a user-friendly online ordering app for Ciel Consulting that supports multiple devices and operating systems	<ul style="list-style-type: none"> <li>- All functional and non-functional requirements are met as per user stories.</li> <li>- The app supports laptops, tablets, and mobile phones globally.</li> <li>- The app functions smoothly on 2G networks or better.</li> <li>- User-friendly interface and seamless user experience.</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Project Sponsor</b> (Final Approval)</li> <li>- <b>Product Owner</b> (Feature Prioritization)</li> <li>- <b>Business Analyst</b> (Requirement Validation)</li> </ul>
<b>Time</b>	.	<ul style="list-style-type: none"> <li>- Project is completed within the planned timeline and in line with the sprint cycle.</li> <li>- Key milestones (requirement gathering, design, development, testing, deployment) are completed as scheduled.</li> <li>- No significant delays or bottlenecks in project execution.</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Project Manager</b> (Overall Timeline)</li> <li>- <b>Scrum Master</b> (Sprint Completion)</li> <li>- <b>Product Owner</b> (Feature Readiness)</li> </ul>
<b>Cost</b>		<ul style="list-style-type: none"> <li>- The project is completed within the approved budget.</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Finance Manager</b> (Budget Approvals)</li> <li>- <b>Project Sponsor</b> (High-level Budget)</li> </ul>

		<ul style="list-style-type: none"> <li>- Budget allocation is well-managed across design, development, testing, infrastructure, and post-launch support.</li> <li>- Minimal cost overruns and adherence to financial plans.</li> </ul>	Decisions) <ul style="list-style-type: none"> <li>- <b>Project Manager</b> (Budget Tracking)</li> </ul>
<b>Quality (Other)</b>		<ul style="list-style-type: none"> <li>- High-quality user experience validated through user testing and QA.</li> <li>- All critical bugs are fixed before release.</li> <li>- The app is secure, reliable, and performs well under various network conditions.</li> <li>- Compliance with legal and data protection requirements.</li> </ul>	<ul style="list-style-type: none"> <li>- <b>QA Lead</b> (Testing &amp; Quality Standards)</li> <li>- <b>Legal/Compliance Team</b> (Data Protection &amp; Regulatory Compliance)</li> <li>- <b>IT Team Lead</b> (Technical Quality &amp; Performance)</li> </ul>
<b>Risk Management (Other)</b>		<ul style="list-style-type: none"> <li>- Identified risks are effectively mitigated.</li> <li>- Contingency plans in place for high-priority risks.</li> <li>- Timely risk assessment and reporting during project phases.</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Risk Manager</b> (Risk Assessment)</li> <li>- <b>Project Manager</b> (Risk Response Planning)</li> <li>- <b>Project Sponsor</b> (Approval for Major Risk Mitigation Steps)</li> </ul>

|