

Project Proposal Template

Fill out each of the sections below with information relevant to your project and add your company's name. Replace the *italicized* information with information relevant to your project.

Feedback Recommendations:

Company Name: Fintech LLC

Project Proposal

Project Scope Statement

- Project Title: Secure and User-Friendly Internet Banking Platform
- Project Sponsor(s):
- CEO of Fintech LLC – Tech-savvy entrepreneur with a strong background in fintech and digital transformation.
- Business Analyst – Co-founder of Fintech LLC. Serial entrepreneur specializing in financial innovation and customer centric solutions.
 - IT Student Project Team Lead
- Business Context for the System:
- An innovative internet bank expands its business online through convenient, secure, and accessible banking services. It expands its outreach remotely to the unbanked population who seek banking anytime unlike traditional physical banks who are challenged to meet such demands. This futuristic outlook minimizes operative expenses to lower overhead costs, elevation of resource utilization to increase business revenue, and scalability.

Project Scope Description:

- Project Goals:
- Global expansion of banking services accessible remotely 24/7
- User-centric design enhances customer satisfaction, user experience, and reliability of repeat customers.
- Robust security implementation to protect user data, financial transactions, and compliance with financial regulations,
- Reduce overhead costs, increase business revenue, and innovative digital tools.
 - Deliverables:
- A functional internet banking platform offers core banking features (account opening, transaction management, bill pay, account monitoring),
- The cloud-based website should demonstrate functionality of its core financial features, personal and business accounts management system, credit and debit cards, loans, and insurance.
- Accessible UX/UI, robust security measures, and implementation of role-based multi-factor authentication.
- Account management allows users to view and manage their account details
- Date Prepared: May 28th, 2024
- Prepared By: IT Student Project Team Lead

Problems/Issues/Opportunities the Proposed System Expected to Solve

Problems	Issues	Opportunities
----------	--------	---------------

Cybersecurity vulnerability	Traditional banking systems can be vulnerable to identity theft, fraud, cyber-attacks, or data breaches due to traditional methods of verification and transaction processing leading to unauthorized access and monetary loss.	<p>Multi-factor authentication: Requires multiple forms of verification to ensure user identity for robust security measures.</p> <p>Implement encryption for data at rest and in transit to protect sensitive information, real-time fraud detection to minimize financial fraud, identity theft, and unauthorized access.</p> <p>Strict data protection, secure transaction processing.</p> <p>Regulatory compliance deters cyber-attacks, and data breaches.</p> <p>Secure Coding Practices to avoid vulnerabilities like SQL injection, XSS, etc.</p> <p>Regular security audits to assess penetration testing.</p> <p>Incident response plan to instantly address security breaches.</p>
-----------------------------	---	---

Project Objectives

Project Objective Name	Project Objective Description (Organizational and departmental goals/objectives addressed by the proposed work)
Expand access and financial inclusion	Organizational goals: Expansion of customer outreach to unbanked populations and underserved geographical regions.

	<p>Departmental goals: Development of features and services remotely catering to a diverse population with diverse banking needs.</p> <p>Objective Description: Provision of online banking platform that caters multiple devices in multiple geographic regions especially the underserved geographic areas where traditional banks lack services.</p>
Ensure security and compliance	<p>Organizational goals: Protection of client data to ensure customer trust in digital banking services.</p> <p>Departmental goals: Implementation of advanced security features in maintaining compliance with financial regulations.</p> <p>Objective Description: Integration of robust security measures i.e., data encryption, real time fraud detection, multi-factor authentication to protect client data and financial transactions. Ensure system adherence to regulatory requirements i.e., GDPR, PSD2, EFTA, BSA, CIP.</p>
Reduce costs, increase financial competence	<p>Organizational goals: Minimize operative expenses augment resource consumption</p> <p>Departmental goals: Automate routine customer services and transaction operations to reduce manual intervention.</p> <p>Objective Description: Implement automated customer support and transactional operations to the online banking platform to reduce operational overhead burdens</p>
Develop comprehensive	<p>Organizational goals: Design a high-level multi-channel customer support system catering diverse user needs or issues</p>

customer support model	<p>Departmental goals: Offer a variety of support channels, leveraging technology for efficiency, ensuring timely and efficient services, and promoting self-service to handle customer issues or queries through digital channels in building trust and fostering long term customer loyalty.</p> <p>Objective Description: Development of a robust platform delivering 24/7 access to remote banking services via live chat, a comprehensive FAQ section, email support, customer help center, and user-friendly well-organized knowledge base for self-service to provide customers with troubleshooting steps.</p>
------------------------	--

Project Deliverables

Project Deliverable Name	Project Deliverable Description (Specific products to be delivered)
UX/UI Design	<p>Products Delivered: Comprehensive exceptionally reliable wireframes, design prototypes, and mockups optimized for performance and accessibility observes modern UX/UI features assure seamless, spontaneous user experience on the web.</p> <p>Features Included: Login Screen, Account Dashboard, Account Management, Transaction Processing, Funds Transfer, Bill Pay, Customer Service, and Customer Support System through web portal.</p>
Security and Compliance	<p>Product Delivered: Integration of comprehensive security measures and compliance frameworks to maintain security standards and compliance regulations.</p>

	<p>Features Included: MFA (Multi Factor Authentication), Data Encryption, Fraud Detection, and Compliance with regulatory bodies.</p>
Customer Support System	<p>Product Delivered: Integration of customer support tools on the web embedded within the internet banking system to support customer satisfaction and efficient resolution of impediments.</p> <p>Features Included: FAQs, Live Chat, Help Center, and Email Support for Queries/impediments.</p>
Back-End Infrastructure	<p>Deliverable Description: Robust back-end infrastructure and databases to support secure, efficient, and scalable functionality of the web platforms.</p> <p>Products Delivered:</p> <ul style="list-style-type: none"> i. Technical Documentation outlines the technical design of the system – SDD document details the system architecture, hardware, and software components used, data flow, and how various parts of the backend interact. • Configuration files (config files) define the parameter preferences applied to the infrastructure components “hardware platform, software platform, OS platform, data storage management, network and communication platform, internet platform, customer and system integration services” (Cynergy Technology, 2023), configure server settings, security protocols, and database connections within the relevant configuration files. <p>Features Included:</p>

	<ul style="list-style-type: none">i. Transaction Processing system validates transaction, account balance updates, integration with payment networks.ii. The User Authentication/Authorization System implements verification MFA of user login credentials and control access to specific features or functions based on user permissions.iii. Data Storage Solutions: a robust DBMS is used to securely store user data, account information, transaction history, and other critical information. For data security, the chosen DBMS should offer sensitive data encryption at rest and in transit.iv. Security Measures include HTTPS communication protocols and intrusion detection and prevention system.
Unit Test & Integration Test Plan	<p>Product Delivered: Unit testing and integration testing document outlines how the system components behaved in unit testing and their interactions with the backend in integration testing. Both plans should be well documented that outlines testing strategy, test case, and expected outcomes. The scope of testing prioritizes user authentication, account management, and transaction processing.</p> <p>Features or Tools Included:</p> <ul style="list-style-type: none">i. User Authentication, unit test and integration test verify the login process validates user credentials and restricts access for invalid login attemptsii. Account Management test confirms account balance retrieval, transaction history display, and profile information updates work as proposed.

	<p>iii. Funds Transfer tests confirm the system accurately calculating the transfer amount, substantiate sufficient funds in the said account, and updates account balance accordingly.</p>
Deployment and Release Plan	<p>Product Delivered: A comprehensive deployment plan and release of cloud-based internet banking system.</p> <p>Deployment Stages:</p> <ul style="list-style-type: none"> i. Development Environment where the system is actively developed and tested ii. Staging Environment replicates the production environment used for final testing and endorsement prior to deployment. iii. Production Environment is the live environment where the users access the internet bank system. <p>Release Criteria: Successful completion of unit testing and integration testing.</p> <p>Features: Timelines, resource allocation, deployment steps, and communication plans for stakeholders to ensure smooth launch of the internet banking system.</p>

Project Acceptance Criteria

Project Acceptance Criteria Name	Project Acceptance Criteria Description (What are the requirements the project must meet in order to be considered complete?)
Functional Completeness	<p>Requirements: User login and authentication</p> <p>Completion Indicator: UX/UI mockups/prototypes should represent UX/UI elements that allow users to interact with login authentication.</p>

	<p>Verification Method: Integration testing verifies the integration of UI with the back-end authentication system.</p>
Comprehensive Customer Support	<p>Requirements: Online help center, live chat, email support, FAQs,</p> <p>Completion Indicator: Implement live chat, FAQs, knowledge base self-service system and ensure their proper functionality (The Stonly Blog, 2022).</p> <p>Verification Method: Test to ensure FAQs for accuracy and clarity. Verify knowledge base is searchable and live chat connects users to agents.</p>
Security and Compliance	<p>Requirements: Security and compliance</p> <p>Completion Indicator: Encryption protocols used for data transmission and storage, secure coding practices to minimize vulnerabilities, access control mechanisms to restrict unauthorized access.</p> <p>Verification Method: Conduct security code review and penetration testing to address security weakness before deployment.</p>
Deployment and Release Plan	<p>Requirements: The system is successfully deployed to the designated production environment.</p> <p>Completion Indicator: According to the deployment plan, the system is deployed on a cloud platform.</p> <p>Verification Method: Smoke testing, system monitoring, and user acceptance testing in production.</p>

Project Exclusions

Project Exclusion Name	Project Exclusion Description (What aspects of the work are outside of the scope of the project?)
------------------------	--

Third-Party Financial Product Integration	The system might not handle loan applications and management. The system may not support international money transfers. The system only supports domestic transfers within a single currency.
Advanced Security Features	Biometric authentication or two factor authentication with hardware tokens may be out of project scope due to limitations.
Mobile App Development	A separate mobile application for the internet bank system is beyond the project scope, the focus at this point is web-based system.
Internationalization	The system to support multiple languages and currencies is out of the scope of this project due to its limitations.

Project Constraints

Project Constraint Name	Project Constraint Description (What are the limitations that the team must work within?)
Time Constraints	The project to be completed with predefined time limit. All phases of design, development, testing, and deployment must be completed within the timeline to avoid delays in project deliverance.
Security Constraints	Robust security measures implementation can be complex and time consuming, may impact timeline or project activities. Limit the use of certain technologies not meeting security standards.
Third-Party Dependencies	The project may depend on third-party vendors for certain components or services e.g., security solutions, payment gateways, or cloud services may impose timeline

	or integration constraints. To avoid delays or additional costs vendor dependencies should be effectively managed.
Budget Limitation	Project pertains fix budget, all development, testing, deployment, and operational activities to be completed within the financial allocation.

Project Assumptions

Project Assumption Name	Project Assumption Description (What information should the team know at this point in the planning?)
System Performance and Security	The architecture and design will meet the expected security and performance benchmarks as planned.
Technology Stack	Project leverages a development team familiar with technology stack (programming languages/frameworks), tools and resources are readily available for the team to ensure efficient development.
Vendor Reliability	Third-party vendor e.g., software licenses, cloud-based services delays or issues may impact deliverables, or timeline. Vendors deliver expected quality product on time,
Third-Party Libraries	The project utilizes pre-built well-maintained third-party libraries to reduce development time and improve code quality.
Project Methodology	

1. Planning:

The project team comes together to set the overall project vision and outline goals. The project team collaboratively maps out a forward plan as soon the project idea is deemed feasible. This is a crucial phase as it lays the foundation for its success. This phase evaluates the idea into manageable parts for their execution. In this phase the primary objective is to translate the initial project idea into well-defined feature constituents called user stories. Functional and non-functional prioritized list of features derived from project requirements is dissected into concise, descriptive user story statements. The user stories are prioritized based on their value to the project. Team may utilize MoSCow prioritization (Agile Business Consortium, n.d.) technique for the project which is sub divided into iterations. An iteration is further divided into timebox to set a timeframe. The iteration/sprint is a short development cycle timeboxed 1-4 weeks. Sprint is an advantage of Agile methodology that provides clear focus for each development cycle and promotes achievable goals (ALTEXSOFT, 2023).

- Internet Bank Vision Statement: Development of Internet Banking System to create user-centric interface and secure online platform to empower account holders for seamless account management services without a schedule or geographical bounds.
- Internet Bank Project Goals: Reduced operational costs, robust security, enhanced user experience economic inclusion of underserved population

2. Requirements Analysis:

This involves several meetings with stakeholders, users, project supervisors, or managers to gather project requirements based on importance and user needs through interviews,

surveys, user observations, open-ended questionnaires, or competitive analysis. The project requirements are analyzed to relevance and quantifiable criteria. Documents gathering ensures the project focuses on delivering the most value able features to their users.

3. Design:

The system design is prepared based on the project requirements. The budget and timelines are defined, and the design team puts together the propositions into a design schema. The test team defines strategies to test each user story as it meets its accomplishments. The design is an advantage of Agile methodology that allows adjustments based on feedback throughout the development process. Daily stand-up meetings and pair programming are put in place to promote collaboration, efficient development, and early detection of potential issues encountered in any part of the designing and integration in the development process. Comprehensive collaborations employ a project from initial phase to development to testing and deployment.

4. Development and Implementation:

The development team builds the functionalities as defined in design. The development team designs in collaboration with the front-end and back-end developers. This is an intensive phase that focuses on coding the system architecture, software, and UI based on the user stories in an iterative developmental approach. The team focuses on creating user-friendly UX/UI, a well-defined system architecture, and the project software. (Cohn, n.d.). Front-end and back-end development for the internet bank involves creation of navigational menu, static web applications, dynamic web applications, and progressive web applications. The back-end development programming language Python and architecture

involves server, databases, back-end web applications (Manturewicz, 2023). API and the middleware development involves CRUD API, SOAP, or RESTful API. Security Implementations are data encryption, authentication, vulnerability testing, authorization. Tools like GitHub and GitLab help in comprehensive Version Control System to protect the source code (ATLASSIAN, n.d.).

5. Testing:

Testing plan moves along as the codes are developed from one iteration to another as defined in the project requirements to match the user stories. Shift-left testing is a great way to push testing in the early stages of development. Automated tests make software development efficient and effective but manual tests, also known as Smoke Tests, are more practical. The team ensures there are no bugs or issues in the newly developed system through four main stages of Agile testing. Incorporating unstructured exploratory testing based on testers domain knowledge uncovers unexpected issues that are not easily discovered in the regular testing operations. In each step of Agile testing, from unit testing to integration testing to system testing and user acceptance testing, the best approach is to combine various comprehensive techniques for a well-rounded testing strategy before the program is deemed safe to release.

6. Deployment Sprint Review & Retrospective:

A sprint review is held at the end of each sprint to display the completed user stories to stakeholders and gather feedback. A sprint retrospective meeting is held to discuss what went well, what challenges arose, and how to improve the process for the next sprint. This allows for continuous improvement based on feedback.

7. Track and Monitor:

Project management tools, burndown charts effectively track progress, identify roadblocks, and manage tasks efficiently for each sprint. Project metrics monitor velocity of work completed per sprint to assess team performance and project strength.

Advantages of Agile Methodology:

Agile as an iterative process offers flexibility and adaptability for changes in requirements based on stakeholder feedback in accommodating adjustments and addressing emerging needs within the sprints. Agile promotes faster delivery on prioritized features in short time-boxed sprints. Daily stand-up meetings to discuss progress, plans for the day, or any impediments, and sprint reviews foster improved communication and stakeholder engagement.

Continuous communication and collaboration within the team and with stakeholders to ensure alignment with the project and address issues promptly. Track key performance indicators (KPIs) such as velocity, sprint burndown, and team satisfaction monitor progress and identify areas for improvement. Finally, user acceptance testing (UAT) and/or stress testing to ensure the product meets all requirements and is ready for deployment.

Celebrating achievements of the team, recognizing the contributions of all team members, and formally closing the project.

High-Level Work Schedule: Project Scope

Description of Work	Assumptions and Constraints
---------------------	-----------------------------

<p>Project Initiation and Planning</p> <p>Description of Work:</p> <ul style="list-style-type: none"> • Define project objectives, scope, and deliverables. • Identify key stakeholders and establish communication channels. • Develop a detailed project plan, including timelines, milestones, and resource allocations. • Conduct a kickoff meeting to align all stakeholders and project team members. 	<p>Assumptions:</p> <ul style="list-style-type: none"> • Stakeholders are readily available and engaged during the initiation phase. • There is a clear understanding of the project’s scope and objectives from all parties involved. • The project team has accessed historical data and insights from similar projects to form planning. <p>Constraints:</p> <ul style="list-style-type: none"> • Budget and timeline for planning activities are limited and must be followed. • The availability of key stakeholders and project team members might be restricted. • The scope definition must be comprehensive yet flexible enough to accommodate minor changes without requiring extensive rework.
---	---

Milestones	Due Dates	Description
Project Start Date	May 30, 2023	Kick-off meeting, project charter, and initial planning.
Gate Review 1: Project Initiation	May 30, 2023	Objective: Review and approve the project scope, objectives, and stakeholder engagement.

	Criteria: -	Approved project charter, clear objectives, and stakeholder list.
Requirement Gathering Completion	June 1, 2023	Complete detailed requirements and functional specifications.
Gate Review 2: Requirement Review	June 2, 2023	Objective: Validate and approve detailed requirements and specifications.
	Criteria: -	Comprehensive requirements document, use cases, and user stories validated and approved by stakeholders.
Development Stage -1	July 1, 2023	Start development focusing on core functionalities and basic architecture.
Gate Review 3: Development Readiness	July 1, 2023	Objective: Ensure development readiness with all necessary preparations in place.
	Criteria: -	Approved system design, development environment set up, initial codebase reviewed.
Stakeholders Feedback on Prototype	July 21, 2023	Deliver a working prototype to stakeholders for feedback and initial testing.
Gate Review 4: Prototype Approval	July 22, 2023	Objective: Obtain stakeholder approval on the prototype and incorporate feedback.
	Criteria: -	Stakeholder feedback collected, prototype functionality aligned with requirements, and identified changes documented.

Prototype Test/Unforeseen Changes	August 20, 2023	Test the prototype thoroughly, address unforeseen issues, and implement necessary changes.
Gate Review 5: Prototype Testing	August 21, 2023	Objective: Validate the prototype's performance and functionality, approve changes and confirm readiness for full-scale development.
	Criteria: -	Successful testing results, resolved critical issues, and updated prototype reflecting feedback and required changes.
Development Stage – 2	August 21, 2023	Continue full-scale development, focus on additional features integrations.
Gate Review 6: Development Completion	September 21, 2023	Objective: Ensure the complete development is ready for testing with all features integrated.
	Criteria: -	Complete and functional codebase, integrated features and no major pending issues.
System Testing & User Acceptance	October 21, 2023	Perform comprehensive system testing and user acceptance testing (UAT).
Gate Review 7: Testing Completion	October 22, 2023	Objective: Validate the system through thorough testing. Confirm readiness for deployment.
	Criteria: -	Passed system tests UAT signoffs and all critical bugs resolved.
Development & Go-Live Preparation	November 1, 2023	Prepare for system deployment including final checks and user training.

Gate Review 8: Go-Live Approval	November 2, 2023	Objective: Approve the system for deployment to production, ensuring all readiness criteria are met.
	Criteria: -	Completed go-live checklist, trained users and confirmed readiness from all stakeholders.
Go-Live and Post Deployment Support	November 15,	Deploy the system to production and provide initial post deployment support.
Gate Review 9: Post-Deployment Review	November 16, 2023	Deploy the system to production and provide initial post deployment support.
	Criteria: -	System operational in production, no critical post-deployment issues and user feedback collected.
Project Closure and Handover	December 1, 2023	Complete project documentation, perform final reviews, hand over the system to the operational team.
Gate Review 10: Project Closure	December 2, 2023	Objective: Finalize the project document lessons learned and officially close the project.
	Criteria: -	Completed project documentation, documented lessons learned, and formal signoff from all stakeholders.

ID	Activity	Resource	Labor Hours	Labor Rate	Labor Total	Material Units	Material Cost	Material Total	Total Cost
1	Project Kickoff meeting to	Project Manager	40	\$100/hr.	4000	0	0	0	4000

	align stakeholder and team members on project milestones								
2	Requirement gathering	Business Analyst	80	\$50/hr.	\$4000	0	0	0	\$4000
3	System design, development of system components, architecture, data structure (UI features, login, dashboard, client interface)	System Architect, System Admin	80	\$100/hr.	\$8000	0	0	0	\$8000
4	Prototype Test	System Architect/ System Admin	20	\$100	\$2000	0	0	0	\$2000

Quality Requirements	
<ul style="list-style-type: none"> For functional accuracy of all banking functions, account management, funds transfer. Payment processing meets the requirement as specified. The system must perform defined banking functions accurately. 100% completion of specified banking functions verified through functional tests to be without errors. Usability of UX/UI must be user-friendly, intuitive, and redundant. User satisfaction survey rating of at least 85% to be achieved post-launch. Data security must be secure from unauthorized access/security breaches and must comply with industry standard security protocols SSL/TLS encryption validated through penetration testing and security audits. The system must pass these tests without major vulnerabilities at a metric of 100%. 	

- System scalability must be able to accommodate user load and transaction volume accordingly without performance concerns. System performance measured through stress testing to evaluate its ability to handle increased load and speed. The system performance stability must remain stable as specified.
- The system must be globally accessible and meet diverse device compatibility. The automated accessibility testing tools WCAG (Web Content Accessibility Guidelines) to analyze website code and content for compliance with web accessibility standards.

Acceptance Criteria

- The system allows users to register their account securely with a valid email address and secure password. Registered users can login to the system with their credentials, MFA an optional but desired feature. The users can view and update their personal information. The system must ensure changes reflect accurately in the database.
- The system should protect user accounts by secure login sessions, implement idle timeout mechanisms, and validate all user input to prevent malicious code injection attempts.
- System seamlessly integrates with external services - payment gateway or banking API.
- The project deliverables are to be reviewed, accepted, and signed off by stakeholders before the project can be considered complete.

Technical Information

Agile Methodology and Statement of Work (SoW)

Agile geared SoW should be dynamic and flexible, accommodating iterative development and continuous improvement. The high-level deliverables of the SoW for the Internet Bank project will include:

1. User Stories and Requirements Gathering

- Creation and prioritization of user stories.
- Continuous engagement with stakeholders to refine requirements.

2. System Design and Prototyping

- Initial design of system architecture and user interface prototypes.
- Feedback loops to refine the design based on stakeholder inputs.

3. Development and Testing Iterations

- Regular sprints focusing on developing and testing incremental features.
- Integration and system testing to ensure each feature works within the broader system.

4. Deployment and Release Management

- Incremental deployments to a staging environment for testing.
- Final deployment to production with post-deployment support.

5. Documentation and Training

- Documentation of system features, user manuals, and technical specifications.
- Training sessions for users and administrators on the new system.

Technology Stack:

The chosen technology stack should support the requirements and goals of the Internet Bank project, focusing on scalability, security, and performance.

1. Front-End Development:

- **Languages:** HTML5, CSS3, JavaScript, or python
- **Frameworks/Libraries:** Vue, React.js or Angular for dynamic and responsive user interfaces.

2. Back-End Development:

- **Language:** Python
- **Framework:** Django or Flask for building robust and secure backend services.

3. Database Management: The optimal choice in RDBMS and NoSQL will be determined based on specific data requirements, performance needs, and simplicity.

- **RDBMS Options:** MariaDB, MySQL, PostgreSQL, Oracle
 - Use cases: Transactional data, user account management, financial records.
- **NoSQL Options:** OrientDB, CouchDB, MongoDB, Cassandra
 - Use cases: High-volume unstructured data, logs, and analytics.

4. Security Practices:

- **Secure Coding:** Input validation, output encoding to prevent injection attacks and cybersecurity vulnerabilities.
- **Authentication & Authorization:** Implement Multi-Factor Authentication (MFA), Role-Based Access Control (RBAC).
- **Encryption:** Secure data in transit using SSL/TLS and data at rest using AES.

Implementation of regulatory compliance in prevention of cyberthreats and cybersecurity vulnerabilities.

System Architecture

A multi-layered architecture ensures separation of concerns, enhances maintainability, scalability, and improves system security posture.

1. Presentation Layer:

- **Components:** User Interface components developed using HTML5, CSS3, Python, or JavaScript.
- **Responsibility:** Displaying data to users and capturing user inputs.
- **Example:** Web pages for account management, fund transfers, transaction history.

2. Business Logic Layer:

- **Components:** Core functionalities and services written in Python.
- **Responsibility:** Processing user requests, executing business rules, and performing transactions.
- **Example:** Handling user authentication, processing loan applications, executing financial transactions.

3. Data Access Layer:

- **Components:** Database interaction modules with chosen database for storing and retrieving data.
- **Responsibility:** Querying and updating the database, ensuring data integrity.
- **Example:** CRUD operations for account information, transaction records.

4. Security Layer:

- **Components:** Security measures like SSL/TLS, encryption libraries, authentication services.

- **Responsibility:** Protecting data, ensuring secure communications, enforcing access controls, ensuring cybersecurity and compliance with regulatory bodies.
- **Example:** Encrypting sensitive user data, secure login processes, data access policies. SSL/TLS, AES, MFA, RBAC, etc.

Key Technical Attributes

To ensure the system meets its operational goals, the following attributes are addressed:

Attributes	Goal	Strategies	Tool
Data Availability	Ensure data is accessible and redundant whenever required to avoid loss and downtime.	Implement database replication, clustering and backup solutions, use distributed databases with automatic failover mechanisms. Implement a cloud-based infrastructure with high availability SLAs.	AWS RDS multi-AZ deployments or Google Cloud Spanner (for managed databases), MariaDB (use galera cluster or replication), OrientDB leverage its distributed architecture and multi master replication for high availability. CouchDB (multi node deployment), couchbase (with flexible database model).
Performance	Deliver responsive and fast user experience.	Optimize queries and use in-memory caching. Implement load balancing and	Redis or Memcached for caching, Nginx for load balancing. MariaDB (optimizing queries). OrientDB (in-

		content delivery networks (CDNs).	memory caching), CouchDB for efficient documents-based storage.
Reliability	Ensure the system operates correctly under normal conditions.	Deploy microservices for fault isolation, automated testing, and monitoring.	Kubernetes for orchestration, Prometheus for monitoring. Jenkins for automated testing and continuous integration. MariaDB (Galaera Cluster for high availability), OrientDB (automatic failover, replication). CouchDB (conflict resolution and data consistency)
Throughput	Handle data processing efficiently, and a high volume of transactions.	Optimize backend services for high concurrency. Use message queues to handle asynchronous tasks.	Apache Kafka or RabbitMQ for message queuing. MariaDB (horizontal scaling, read replicas), CouchDB (distributed processing). OrientDB (horizontal scaling).
Maintainability	Facilitate easy updates, system maintenance and updates.	Employ modular design and continuous integration/continuous development (CI/CD). Maintain comprehensive	Jenkins for CI/CD, Docker for containerization, Swagger for API documentation. MariaDB (for ease of maintenance with PhpMyAdmin), OrientDB – schema flexibility. CouchDB – easy replication.

		documentation and coding standards.	
Monitoring	Ensure system health, performance, and security to ensure the system meets its SLAs and performs optimally.	Comprehensive monitoring alerting mechanism, analytics, and visualization.	Prometheus (functionality collects and stores metrics, powerful query language, works well with Grafana for visualization and Kubernetes for container metrics). Grafana (functionality provides dashboards and graphs for monitoring and analyzing data, integrates with various data sources e.g., Prometheus, Elasticsearch, and InfluxDB), ELK Stack (Elasticsearch- search & analytics engine, logstash- server-side data processing pipeline for data collection, transformation & storage, kibana- data visualization tool) for centralized logging and analysis. Datadog, Zabbix, Nagios (functionality- monitors system, network, & infrastructure with alerting capabilities, integrate with wide range of plugins & third-party tools), New

			<p>Relic (functionality-application performance monitoring real-time data & analytics, integration supports various languages and frameworks), Splunk (cloud & on-premises) For system health monitoring and alerting.</p> <p>Cloud-Specific Monitoring: AWS CloudWatch, Azure Monitor for detailed cloud resource monitoring.</p>
Management	<p>Facilitate efficient management of infrastructure and services.</p> <p>Low level components servers, storage, networking devices, Cloud native services, OSs</p>	<p>Automation, configuration, management, orchestration.</p>	<p>Infrastructure as Code (IAC)</p> <p>Terraform (day 0, 1), Puppet (day1, 2, help manage resources in Kubernetes).</p> <p>Kubernetes (functionality- manages containerized applications across a cluster of machines, provides automation for deployment, scaling, and operations. Integration works with Docker and other container runtimes, integrates with CI/CD pipelines).</p> <p>Ansible (simple ease of use, configure systems, deploy software, orchestrate advance workflow, supports application deployment, system</p>

	application-level configurations.		updates, etc.), Docker (containerization platform), AWS CloudFormation, Azure Resource Manager (Infrastructure as Code, services for managing resources, deep integration with their respective cloud ecosystems).
--	-----------------------------------	--	--

Tools for Measuring Metrics and KPIs:

Prometheus and Grafana: Real-time monitoring and visualization of metrics.

Nagios: Alerting and monitoring system health.

New Relic: Application performance monitoring APM.

Elk Stack: Centralized logging and log analysis.

Jenkins: Tracking deployment frequency and building success rates.

SonarQube: Measure code quality and maintainability.

Agreement Information

- This agreement outlines to maintain shared responsibility, timely delivery, iterative development, SoW, and the commitment of all the team members involved in the SoW for development of the internet bank system.
- It sets clear expectations, ensures accountability, facilitates smooth and effective collaboration, and maintains timelines throughout the development process.
- It is acknowledged the project allows for adjustments or changes incorporated into subsequent iterations based on evolving requirements and/or feedback.

- **Date of the Agreement:**

The agreement clearly states the date it was entered into by the parties involved. This establishes the timeline for the subsequent activities and obligations.

- **Parties Included in the Agreement:**

- **Client:** The party commissioning the development of the internet bank system.
- **Development Team:** The party responsible for developing and delivering the internet bank system.
- **Stakeholders:** Other relevant parties involved in the project, i.e., financial institutions, regulatory bodies, or subcontractors.

High-Level Information About the Services to be Provided:

- The service description section provides an overview of the services to be provided.
 - Development of an internet bank system includes front-end & back-end development, database setup, and integration with third-party services.
 - Implementation of secure coding practices, input validation, user authentication, encryption, and authorization.
 - Deployment of the system on a reliable and scalable infrastructure.
 - Ongoing maintenance and support services post-deployment.

Cancellation Policy:

- **Conditions for Cancellation:** This section clearly defines the conditions under which either party can terminate the agreement.
 - Breach of contract by either party.
 - Failure to meet agreed-upon milestones or deliverables.

- Significant changes in project scope or requirements that cannot be mutually agreed upon.
- **Notice Period:** This section specifies a 30-day notice period is required for cancellation, allowing adequate time for the other party to respond or take necessary actions.
- **Compensation and Penalties:**
 - Refund for payments made in advance.
 - Payment for work completed up to the date of cancellation.
 - Penalties for premature termination may be applicable.
- **Post-Cancellation Obligations:** Detailed obligations persist after cancellation, i.e., returning confidential information, transferring intellectual property, and/or providing transition support.

REFERENCES

Agile Business Consortium. (n.d.). *MoSCoW Prioritization*.

<https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html>

ALTEXSOFT. (2023, May 10). *Iterative Process in Agile*.

<https://www.altexsoft.com/blog/iterative-process-agile/>

ATLASSIAN. (n.d.). *What is version Control?* [https://www.atlassian.com/git/tutorials/what-is-version-](https://www.atlassian.com/git/tutorials/what-is-version-control#:~:text=Version%20control%20software%20keeps%20track,disruption%20to%20all%20team%20members)

[control#:~:text=Version%20control%20software%20keeps%20track,disruption%20to%20all%20team%20members](https://www.atlassian.com/git/tutorials/what-is-version-control#:~:text=Version%20control%20software%20keeps%20track,disruption%20to%20all%20team%20members)

Cynergy Technology. (2023, August 31). *7 Components of IT Infrastructure*.

[https://www.cynergytech.com/stories/7-components-of-it-infrastructure-definitions-](https://www.cynergytech.com/stories/7-components-of-it-infrastructure-definitions-features/#:~:text=An%20IT%20infrastructure's%20seven,customer%20and%20system%20integration%20services.)

[features/#:~:text=An%20IT%20infrastructure's%20seven,customer%20and%20system%20integration%20services.](https://www.cynergytech.com/stories/7-components-of-it-infrastructure-definitions-features/#:~:text=An%20IT%20infrastructure's%20seven,customer%20and%20system%20integration%20services.)

Cohn, M. (n.d.). *Incorporating UI Design in Agile Sprints*. Mountain Goat SOFTWARE.

<https://www.mountaingoatsoftware.com/blog/incorporating-ui-design-in-scrum-sprints>

Davis, K. D. (2024, May 19). *Project Closure Process*. UPGrad Knowledgehut.

<https://www.knowledgehut.com/blog/project-management/project-closure>

Eby, K. (2017, February 15). *Agile Methodology What's the Difference?* Smartsheet.

<https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban> FFIEC (n.d.). *FEDERAL*

FINICIAL INSTITUTIONS EXAMINATION COUNCIL GUIDANCE ON ELECTONIC

FINANCIAL SERVICES AND CONSUMER COMPLIANCE. <https://www.ffiec.gov/pdf/efs.pdf>

Hostinger Tutorials. (n.d.). *What is New Relic*. [https://www.hostinger.in/tutorials/what-is-new-](https://www.hostinger.in/tutorials/what-is-new-relic#:~:text=New%20Relic%20is%20a%20real,app%20that%20tracks%20their%20performance.)

[relic#:~:text=New%20Relic%20is%20a%20real,app%20that%20tracks%20their%20performance.](https://www.hostinger.in/tutorials/what-is-new-relic#:~:text=New%20Relic%20is%20a%20real,app%20that%20tracks%20their%20performance.)

Manturewicz, M. (2023, May 4). *25 Backend Tools to facilitate web development*. Codilime.

<https://codilime.com/blog/backend-tools/#:~:text=It%20connects%20the%20frontend%2C%20that,frontend%20is%20the%20client%20side>

Melashich, S. (2024, May 2). *Top Security Practices for Neobanking You Need to Know*. Agile.

<https://agilie.com/blog/top-security-practices-for-neobanking-you-need-to-know>

OCO Online Consortium of Oklahoma. (n.d.). *Agile Project Managment*.

<https://open.ocolearnok.org/informationssystem/chapter/chapter-13-agile-project-management/>

Project Management Institute. (2003, May 25). *Project gates*.

<https://www.pmi.org/learning/library/contemporary-gate-philosophy-implemented-outcome-7786>

The Stonly Blog. (2022, February 4). *How to Build a Help Center? The Complete Step-by-Step*

Guide <https://stonly.com/blog/build-help-center/>

The Stonly Blog. (2022, August 10). *12 Best Knowledge Base Software*.

<https://stonly.com/blog/knowledge-base-software/>

Test Rail. (n.d.). *Agile QA Best Practices*. <https://www.testrail.com/blog/agile-qa-best-practices/>

Testsigma Engineering Team. (2018, June 18). *Unit Testing / What it is, How it Works, Types &*

Top Benefits. <https://testsigma.com/blog/unit-testing/>

University of Mississippi - Ole Miss IT LABS. (n.d.). *The J2EE Tutorial*.

<https://john.cs.olemiss.edu/~hcc/distObj/notes/j2eetutorial/index.html>