Project Rollout, Postmortem, and Wrap-Up

David Crane

UofPhx

BSA/425

Bill Schuyler

10/16/2023

Table of Contents

Project Implementation Plan

Replace the *italicized* information in columns 2 through 6 with information relevant to your project.

Project Name: CAPS LOCK Bank Digital Banking Platform Implementation

Implementation Phase: Deployment & Go-Live

Project Plan

Task	Activity Name	Resource	Schedule Start Date	Schedule Finish Date	Schedule Comments
Prepare for Implementation	Formulate the implementation team	CIO/Project Sponsor	11/13/2023	11/17/2023	Clearly define and communicate each team member's roles, responsibilities, and key deliverables.
Prepare for Implementation	Acquire server hardware for all environments	IT Manager/CIO	11/18/2023	12/2/2023	Confirm hardware specifications with vendors and ensure timely delivery to avoid project delays.
Prepare for Implementation	Acquire client hardware for all environments	IT Manager/CIO	11/18/2023	12/2/2023	Validate hardware-software compatibility and ensure all units meet the necessary specifications.
Prepare for Implementation	Procure Cloud Computing Services and software subscriptions	IT Manager/CIO	12/3/2023	12/17/2023	Ensure negotiated terms align with project budget and verify compliance with data protection regulations.
Prepare for Implementation	Secure on-premise software applications	IT Manager/CIO	12/18/2023	1/1/2024	Confirm software licenses are valid for the project duration and check installation prerequisites to avoid setup delays.
Prepare for Implementation	Establish user IDs and groups for on-premise technology	System Admin	1/2/2024	1/17/2024	Establish secure user IDs and groups, ensuring appropriate access levels are granted in line with role requirements.

Task	Activity Name	Resource	Schedule Start Date	Schedule Finish Date	Schedule Comments
Prepare for Implementation	Develop security infrastructure for all environments	System Admin	1/18/2024	2/1/2024	Ensure the developed security infrastructure adheres to organizational and industry cybersecurity standards.
Prepare Test Environment	Set up test/QA servers and software	System Admin	2/2/2024	2/6/2024	Confirm the test/QA environment is isolated from production and secured against unauthorized access.
Prepare Test Environment	Set up test/QA client machines and software	System Admin	2/7/2024	2/11/2024	Ensure client machines are configured accurately and verify access for relevant testing teams.
Prepare Test Environment	Configure test/QA environment	System Admin	2/12/2024	2/16/2024	Confirm the test/QA environment mirrors the production environment to ensure accurate testing results.
Prepare Production Environment	Set up production servers and software	System Admin	2/17/2024	2/21/2024	Ensure servers and software are optimized for peak performance and secured against potential threats.
Prepare Production Environment	Set up production client machines and software	System Admin	2/22/2024	2/26/2024	Validate configurations of all client units and ensure consistency across the production environment.
Prepare Production Environment	Configure production environment and integrate with cloud service architecture	System Admin	2/27/2024	3/3/2024	Confirm configurations with the cloud service provider and ensure seamless integration with the production environment.
Data Conversions	Execute data conversions and loading	Data Specialists	3/4/2024	3/18/2024	Ensure data integrity post-conversion and check for consistency

Task	Activity Name	Resource	Schedule Start Date	Schedule Finish Date	Schedule Comments
	into the test/QA environment				across the test/QA environment.
Data Conversions	Execute data conversions and loading into the production environment	Data Specialists	3/19/2024	4/2/2024	Validate data synchronization and accuracy, ensuring no discrepancies post-conversion.
Documentation	Validate all documentation applicable to implementation is available	Documentation Specialists/Technical Writers	4/3/2024	4/17/2024	Confirm all documentation is comprehensive, clear, and easily accessible to all relevant stakeholders.
Training	Conduct administrator training	Trainers	4/18/2024	4/22/2024	Ensure administrators demonstrate understanding and capability in managing the new platform.
Training	Conduct support training	Trainers	4/23/2024	4/27/2024	Confirm the support team is fully trained and prepared to address user issues effectively.
Training	Conduct end-user training	Trainers	4/28/2024	5/2/2024	Ensure users demonstrate comfort and basic proficiency with the new platform.
Go-Live	Initiate Production Go-live	CIO/Project Sponsor	5/3/2024	5/3/2024	Closely monitor system performance, address any issues promptly, and ensure support is readily available for all users.

Post-Mortem

Use the table to list the things that went well during the completion of this 5-week project and the things that didn't go well.

Date: 10/16/2023

Project Manager (your name): David Crane

Project Name: CAPS LOCK Bank Digital Platform Implementation

5 Things that Went Well During the Project	5 Things that Could Have Been Done Better
Comprehensive Project Planning	Real-world Testing
Detailed Technical Specifications	Resource Allocation
Hybrid Project Methodology	Risk Management
Vendor Selection	Stakeholder Engagement
Milestone Planning	Post-Implementation Support

Below, discuss all 10 things from the list above in detail. As a project manager, or participant, what processes might you put in place to ensure the same things go well on future projects you're involved in? What processes can you put in place to improve the not-so-great things on future projects? Be specific.

Approaches That Worked Well

- **Comprehensive Project Planning:** The meticulous planning ensured clear objectives, scope, and deliverables. A specific approach that worked well here was the detailed breakdown of each phase of the project, ensuring clarity and direction.
- **Detailed Technical Specifications:** The clear and well defined technical details kept the development and implementation phases direct and straightforward. Starting technical feasibility comparisons early, combined with implementing a thorough test plan was an efficient approach.
- **Hybrid Project Methodology:** The adoption of a hybrid project methodology ensured a blend of flexibility and structured foundational steps. This was particularly effective in maintaining a balance between adhering to financial compliance and allowing for iterative development.
- **Vendor Selection:** The structured vendor selection process, involving a rigorous approach to the vetting criteria, ensured that the project was not hindered by inadequate 3rd parties.

- **Milestone Planning:** Clear milestones and due dates provided a structured timeline for the project. This was crucial in ensuring timely delivery and keeping the project on track.

Improving Future Projects

- **Real-world Testing:** The lack of real-world testing limited the ability to gather real-world data and feedback to incorporate. An innovative approach to improve this in future projects could be to simulate user journeys and utilize feedback from a diverse group of user personas.
- **Resource Allocation:** A more explicit resource allocation plan, including a more broken down detailed purchase budget and risk/contingency plan cost estimates could have been beneficial. Future projects could implement a detailed resource management plan, ensuring clear budget allocation, personnel assignment, and contingency planning for resource-related risks.
- **Risk Management:** A more detailed risk management strategy could be beneficial. Involving all stakeholders in a detailed risk assessment phase to identify potential risks and develop mitigation and contingency plans can enhance this in future projects.
- **Stakeholder Engagement:** Enhancing stakeholder engagement throughout the project lifecycle is crucial. Implementing regular stakeholder update meetings and feedback sessions can enhance engagement and ensure alignment with stakeholder expectations and needs.
- **Post-Implementation Support:** Ensuring the longevity and success of a project post-implementation is crucial. Future projects should include a detailed post-implementation support plan, ensuring a clear support structure, training for support personnel, and a structured feedback process for continuous improvement post-launch.

Project Proposal

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.

Company Name: CAPS LOCK Bank

Project Proposal

Project Scope Statement

- **Project Title:** Internet Banking Digital Platform Implementation
- **Project Sponsor(s):**
  - **Executive Sponsor**
  - **Chief Technology Officer**
  - **Investor Relations Manager**
  - **Digital Strategy Lead** (self)
- **Business Context for the System:** We are introducing an Internet bank to offer a full suite of digital financial solutions. This endeavor aims to elevate the user experience and simplify financial management, reflecting the demands of contemporary banking and meeting the evolving needs of our customers.
- **Project Scope Description:**
  - **Goal(s):**
    - Launch a comprehensive digital banking platform.
    - Ensure 24/7 platform availability with the highest security standards.
    - Facilitate financial management for users across various services.
  - **Deliverables:** A user-friendly, secure, and scalable digital banking platform.
  - **Features and Functions:** Digital accounts, card management, loan processing, insurance services, investment management, tax services.
- **Date Prepared:** September 18, 2023 **Updated:** 10/16/2023
- **Prepared By:** David Crane

Problems/Issues/Opportunities the Proposed System Expected to Solve

Problems	Issues	Opportunities
Transaction Speed	Slow processing of traditional banking transactions	Instantaneous online transactions
Physical Constraints	Traditional banking requires in-person visits & paperwork	Entirely digital, no physical presence needed

Limited Accessibility	Limited access to banking outside business hours	24/7 global banking access
Security Concerns	Increased cyber threats to online banking platforms	Multi-level authentication & robust security measures

Project Objectives

Project Objective Name	Project Objective Description (Organizational and departmental goals/objectives addressed by the proposed work)
Scalable Digital Infrastructure	Build a platform that can handle a high influx of users without service degradation.
24/7 Availability	Ensure platform uptime all day, every day, with minimal outages or interruptions.
Robust Security	Implement top-tier security measures, encryption, and multi-level user authentication.
User-friendly Interface	Design an intuitive user interface for ease of banking operations.

Project Deliverables

Project Deliverable Name	Project Deliverable Description (Specific products to be delivered)
Requirements	Define technical and functional requirements for the internet bank, ensuring coverage for all services (accounts, cards, loans, insurance, investments, and tax services).
Analysis	In-depth review of technical requirements, risk assessments, and user expectations to ensure the system meets both business and user needs.

Development	Construct the core systems for each service, emphasizing security, scalability, and availability. Develop test cases for each functionality.
Prototype	Launch a beta version of the platform, gather user feedback, and make necessary adjustments. Ensure all functionalities are intuitive and meet users' expectations.
Implementation	Full deployment of the digital banking platform. Train end-users, onboard customers, and monitor system performance in real-time. Ensure 24/7 support.

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?)
Performance	The system must handle projected user loads without significant lag, errors, or system crashes.
Security	The platform should meet industry-standard security protocols, with successful penetration testing results, ensuring user data protection and transaction security.
Functionality	All banking modules (digital accounts, card management, loan processing, insurance, investment, and tax services) must be operational, user-friendly, and perform their core functions effectively.
User Experience	The platform should offer an intuitive and seamless user experience, with a maximum of three clicks to any essential function.
Compatibility	The platform should be compatible across multiple devices (desktop, mobile, tablet) and main browsers (e.g., Chrome, Firefox, Safari).
Uptime	The platform should maintain a minimum of 99.9% uptime, excluding scheduled maintenance.

Project Exclusions

Project Exclusion Name	Project Exclusion Description (What aspects of the work are outside of the scope of the project?)
Physical Infrastructure	The project will solely focus on the digital platform; there will be no establishment of physical bank branches or ATMs.
External Integrations	Integrations with third-party software or platforms not included in the initial scope are excluded. Future integrations may require additional resources and time.

Project Constraints

Project Constraint Name	Project Constraint Description (What are the limitations that the team must work within?)
Budget	Financial constraints dictate the resources available. Efficient allocation is essential to ensure every facet of the project is adequately funded without overspending.
Time	The timeline is strict, aiming to go live within a year. This constraint mandates effective time management and prioritization.
Resource Limitations	While aiming for rapid deployment, the project must also grapple with available personnel and technology resources, ensuring neither is overstretched.

Project Assumptions

Project Assumption Name	Project Assumption Description (What information should the team know at this point in the planning?)
Team Availability	It's presumed that essential team members, both from the managerial and technical side, will be present and active throughout the project's lifespan.
Team Capability	The team possesses the expertise and skills necessary to bring the project to fruition, from planning to execution.

Vendor Reliability	The external vendors, be it for software or other services, will honor their commitments in terms of deliverables and timelines.
User Onboarding	The target audience possesses the foundational digital skills required to navigate and benefit from the platform.
Regulatory Compliance	All platform operations and features will align with banking regulations, ensuring legal compliance and building user trust.

Project Methodology
We will adopt a hybrid approach combining Agile principles with foundational steps essential for financial platforms. **1. Requirements Gathering:** ● Define technical and user specifications. ● Understand regulatory requirements for online banking. **2. System Design:** ● Outline the technical framework and security protocols. **3. Agile Development Cycles:** ● Iterative development focused on key banking features. ● Regular reviews and backlog adjustments based on feedback. **4. Infrastructure and Testing:** ● Set up essential infrastructure. ● Integrate continuous testing to ensure system robustness. **5. User Acceptance Testing (UAT):** ● Engage select users to validate platform functionality. **6. Deployment:** ● Launch the platform to the public. ● Monitor initial launch for unexpected issues. **7. Feedback & Continuous Improvement:** ● Embrace user feedback. ● Implement refinements in future development cycles.

Advantages of this methodology:

- Combines the flexibility of Agile with the rigidity needed for financial platforms.
- Enables adaptive changes based on real-time feedback.
- Ensures rigorous compliance checks without compromising on user experience.
- Facilitates strong team and stakeholder communication.

High-Level Work Schedule: Project Scope

Description of Work	Assumptions and Constraints
• Utilize Agile methodology to implement the digital banking platform. • Set up a robust and secure cloud-based infrastructure for the platform. • Develop and integrate key features including digital accounts, card management, loan processing, insurance portal, investment platform, and tax services. • Ensure full integration of the platform's modules for seamless user experience.	• Team works a standard 8-hour day, 5 days a week. • Changes in regulatory compliance might influence platform features and timeline. • Task delegation based on team members' expertise to ensure optimal results.

Milestones	Due Dates
Project Start Date	10/1/2023
Initial Development & Platform Setup	11/30/2023
Launch of Beta Version for Internal Testing	3/1/2024
Stakeholders Feedback on Beta Version	3/20/2024
Refinement & Full Integration of Banking Services	5/15/2024
Public Launch of the Internet Banking Platform	9/1/2024
Final Testing and Adjustments	9/20/2024

Unforeseen challenges or modifications	May add between 15-75 days (estimate) to timeline

ID	Activity	Resource	Labor Hours	Labor Rate	Labor Total	Material Units	Material Cost	Material Total	Total Cost
1	Web Application Development	Software Developer	800	$50	$40,000	10	$100	$1,000	$41,000
2	Backend Server Setup	IT Specialist	500	$60	$30,000	5	$500	$2,500	$32,500
3	Multi-layer Security Implementation	Cybersecurity Expert	600	$70	$42,000	2	$300	$600	$42,600
4	User Interface Design	UI/UX Designer	400	$55	$22,000	5	$150	$750	$22,750
5	Database Infrastructure	Database Admin	500	$65	$32,500	3	$600	$1,800	$34,300
6	Maintenance & Support (Quarterly)	Tech Support	800	$45	$36,000	0	$0	$0	$36,000
7	Regulatory Compliance & Auditing	Compliance Officer & Auditors	600	$80	$48,000	5	$200	$1,000	$49,000
8	Platform Training	Training Specialist	400	$50	$20,000	10	$100	$1,000	$21,000

Quality Requirements

(What specific, measurable requirements must the project meet?)

- **Scalability**: The system must be capable of handling anticipated user growth seamlessly without affecting service quality.
- **Availability**: Ensure 24/7 reliable access with optimal performance.
- **Security**: Incorporate multi-layered authentication and industry-standard encryption.
- **Manageability**: The platform should be easy to update, manage, and support.
- **User Experience**: Intuitive, user-friendly interface accessible on various devices.
- **Comprehensive Services**: All financial services must be integrated and easy to navigate.
- **Backup & Redundancy**: Ensure continuous service with data backup and system redundancies.

Acceptance Criteria
(What specific results or functional requirements should the work yield to be considered completed and approved?) ● Platform must be successfully deployed to a small test group for initial feedback. ● Testing must encompass both functional (how it works) and non-functional (how well it works) aspects. ● All core banking modules, such as digital accounts, card management, and loan processing, must be fully functional and integrated. ● Confirmation that all built-in automated processes are operational and have undergone rigorous testing. ● Ensure the platform is compliant with the relevant regulatory guidelines.

Technical Information
● Development will utilize the Agile methodology for iterative design and improvements. ● Continuous testing will be implemented throughout the build phase to ensure optimal performance and reliability. ● The platform will utilize cloud-based infrastructure to ensure scalability, redundancy, and high availability. ● Encryption standards and security protocols will adhere to the highest industry standards.

Agreement Information
All teams are dedicated to their tasks from the project's start to its end. This agreement ensures everyone collaborates effectively. The goal is to keep the project on track and on time. While we can make adjustments along the way, any changes will align with the SoW (Scope of Work) to maintain clarity in the project's goals.

Project Plan

Fill out each of the sections below with information relevant to your project. Be sure to include the company name associated with your project.

Company Name: CAPS LOCK Bank

Updated: 10/2/2023, 10/16/2023

Network Technology Recommendations

Network Technology Selection Criteria

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value *(e.g., 3 – Excellent, 2 – Good, 1 – Acceptable)*
Scalability	Can handle high growth and a large influx of new users without affecting service levels for existing users.	3 - Excellent
Availability	Ensures robust, consistent, and reliable access 24/7.	3 - Excellent
Security	Should have industry-accepted security practices and support multi-level authentication.	3 - Excellent
Manageability	Must be easy to manage, support, and update.	2 - Good

Network Technology Recommendation

Recommended Network Technologies	Description	Benefits	Aggregate Selection Criteria Score
Software-Defined Networking (SDN)	Highly scalable, allows for easy management and effective traffic handling.	Aligns with key factors like scalability, manageability, and availability.	11
Zero Trust Network Architecture	Security model that requires strict identity verification for every person and device	Enhances security by enforcing multi-level authentication and least-privilege access.	11

	trying to access resources on a private network.		
Enhanced Security Measures	Implementation of advanced fraud detection technologies, with AI and machine learning capabilities.	Real-time prevention of suspicious activities, strengthening the security of online transactions.	11

Network Technology Vendor Selection Criteria

(*Third-party technology provider*)

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value (Weighting in Points)
Reliability	Proven uptime and reliability.	3 - Excellent
Customer Support	24/7 customer service and technical support.	2 - Good
Scalability	Vendor can adapt to high growth scenarios	2 - Good
Compatibility	Integration with existing infrastructure	2 - Good

Network Technology Recommended Vendors

Vendor Name	Vendor Strengths	Vendor Weaknesses	Products/Services Provided to Project	Aggregate Selection Criteria Score
Cisco (Cisco Systems, Inc., 2023)	Highly reliable, excellent customer support.	Cost	Cisco switches, SDN controllers	9
Palo Alto Networks (Palo Alto Networks, 2023)	Excellent in network security solutions.	Complexity in some setups.	Next-Gen Firewalls, Zero Trust architecture solutions.	8

Network Technology Deployment Challenges

Deployment Challenge *(short name to ID the challenge)*	**Deployment Challenge Description** *(What obstacles can potentially complicate or delay deployment of technology, and affect the project timeline?)*
Initial Setup	Coordinating all initial configurations and setups may be complex and time-consuming.
Staff Training	New technologies may require specialized skills that existing team members do not have.
Risk Mitigation	Handling technology integration failures during initial setup
Rollback Plan	Plan for reverting changes in case of critical failures

Technology Adoption Methods

Method Name *(short name to ID the method)*	**Method Description** *(Summarize the process for adopting the technology.)*
Phased Rollout	Roll out new technologies in phases, monitor performance
Continuous Monitoring	Continuous monitoring for network stability and security
Backup Plan	Contingency plan for unexpected challenges during phased rollout

Cost/Benefit Considerations

Benefits	Costs	Considerations
Real-time Traffic Management	Hardware Investment, Ongoing Maintenance and Operation Costs	Bandwidth Requirements, Impact on real-time transactions
High Redundancy: Minimized downtime	Software Licensing: Cost of network management tools	Latency: Impact on real-time transactions
Enhanced Trust/Regulatory Compliance	Increased Investment	Staff Training and Integration

Database System Recommendation

Database System Selection Criteria

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value (Weighting in Points)
Scalability	Ability to handle a growing amount of data and transactions seamlessly.	3 - Excellent
Security	Robust encryption, data masking, and multi-level authentication.	3 - Excellent
Performance	Quick data retrieval and low latency.	2 - Good
Manageability	Ease of management, support, and updates.	2 - Good

Database System Recommendation

Recommended Database System	Description	Benefits	Aggregate Selection Criteria Score
PostgreSQL (McAllister, 2023)	Open-source, ACID-compliant, and offers robust performance.	High scalability, strong community and vendor support, advanced security features.	10
NoSQL [MongoDB] (MongoDB, Inc., 2023)	Ideal for handling large volumes of unstructured data, excellent for data analytics.	High performance, flexible schema, easy to scale.	8
Database System Diversity	Adoption of a polyglot persistence architecture, utilizing both SQL and NoSQL databases.	Enhanced flexibility and scalability, efficient handling of diverse data types and large volumes.	8

Database System Vendor Selection Criteria

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value (Weighting in Points)
Vendor Reliability	Proven track record in the industry.	3 - Excellent
Support & Maintenance	Regular software updates and 24/7 support.	2 - Good
Scalability	How well the vendor's solutions can adapt to growth	2 - Good
Compatibility	Compatibility with existing or planned system architecture	2 - Good

Database System Recommended Vendors

Vendor Name	Vendor Strengths	Vendor Weaknesses	Products/Services Provided to Project	Aggregate Selection Criteria Score
Oracle Corporation (Oracle, 2023)	Extensive feature set, high reliability.	High costs.	Oracle PostgreSQL Database Service	8
MongoDB Inc. (MongoDB, Inc., 2023)	Scalability, strong community support.	Learning curve.	MongoDB Atlas	7

Database System Deployment Challenges

Deployment Challenge	Deployment Challenge Description
Initial Setup	Configuring the initial settings and parameters can be time-consuming.
Skill Gap	Ensuring the team has the necessary skills to manage the new database system.
Risk Mitigation	Handling database integration issues during initial setup.
Rollback Plan	Procedures to revert changes in case of critical issues.

Technology Adoption Methods

Method Name	Method Description
Phased Migration	Implement the new database system incrementally, migrating data in stages.
Continuous Monitoring	Monitor performance metrics and error logs from day one.
Backup Plan	Contingency plan in case of unexpected issues during migration.

Cost/Benefit Considerations

Benefits	Costs	Considerations
Latency: Impact on real-time transactions	Storage Costs: Expenses for data storage solutions	Backup and Recovery: Planning and costs
High Availability Setup	Backup and Recovery Costs	Availability Level, Data Integrity
Data Handling Efficiency	Integration Complexity	Scalability Planning

Software Application Recommendations

Software Application Selection Criteria

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value (Weighting in Points)
User Experience (UX)	The ease of use and intuitiveness of the software.	3 - Excellent
Scalability	Ability to grow and adapt to an increasing number of users.	3 - Excellent
Security	Measures to protect customer data and transactions.	3 - Excellent
Cost	Total expense including initial purchase, licensing, and ongoing maintenance.	2 - Good

Software Application Recommendation

Recommended Software Application	Description	Benefits	Aggregate Selection Criteria Score
Temenos T24 Transact (Temenos Headquarters SA, 2023)	Core banking solution with modular components for scalability.	High security and an intuitive user interface.	11
Finastra Fusion Essence (Finastra, 2023)	Cloud-based core banking ideal for smaller banks.	User-friendly with moderate setup costs.	10
Optimization of User Experience	Engagement with specialized UX design firms for a responsive, customizable platform.	An intuitive user interface that meets evolving customer expectations across various devices.	10

Software Application Vendor Selection Criteria

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value (Weighting in Points)
Reputation	Public and industry reputation of the vendor.	3 - Excellent
Support and Maintenance	Level of support and maintenance services.	2 - Good
Scalability	How well the vendor's solutions can adapt to growth	2 - Good
Compatibility	Compatibility with planned system architecture	2 - Good

Software Application Recommended Vendors

Vendor Name	Vendor Strengths	Vendor Weaknesses	Products/Services Provided to Project	Aggregate Selection Criteria Score
Temenos (Temenos Headquarters SA, 2023)	Excellent reputation, extensive support network.	Higher initial setup cost.	Temenos T24 Transact	8
Finastra (Finastra, 2023)	Good reputation, reasonable pricing.	Less extensive support network.	Finastra Fusion Essence	8

Software Application Deployment Challenges

Deployment Challenge	Deployment Challenge Description
Data Migration	Challenge of securely and accurately transferring existing data into the new system
User Training	Need to educate staff and potentially end-users on how to use the new software effectively
Risk Mitigation	Contingency plan for failed software integration during deployment

Technology Adoption Methods

Method Name	Method Description
Parallel Run	Running the new system in parallel with dummy data before fully committing.
Incremental Deployment	Introducing the new system gradually, module by module, to ensure each part works as expected before full-scale deployment.
Backup Plan	Contingency plan in case of unexpected issues during implementation

Cost/Benefit Considerations

Benefits	Costs	Considerations

API Flexibility: Easy integration with other services	Customization, Ongoing Maintenance and Operation Costs	Integration Complexity, Data Integrity
Customer Retention	User Training: Costs for educating staff/users	Version Upgrades: Impact of future software updates
Increased Accessibility	Design Expenses/Ongoing Improvement Costs	Device Compatibility

Cloud Services Recommendations

Cloud Services Selection Criteria

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value (Weighting in Points)
Scalability	Ability to scale resources up or down according to demand.	3 - Excellent
Availability	99.99% uptime with geographically distributed data centers.	3 - Excellent
Security	Robust encryption, DDoS mitigation, compliance with industry standards.	3 - Excellent
Manageability	Easy to manage with robust tools for resource management.	2 - Good

Cloud Services Recommendation

Recommended Software Application	Description	Benefits	Aggregate Selection Criteria Score
Amazon Web Services (AWS) (Amazon Web Services, Inc., 2023)	Secure cloud platform offering a wide range of services.	High availability, robust security, and scalability.	11

Microsoft Azure (Microsoft, 2023)	Flexible tools for building, managing, and deploying applications	Good scalability, integration capabilities, and security features.	10

Cloud Services Vendor Selection Criteria

Selection Criteria Name	Selection Criteria Description	Selection Criteria Value (Weighting in Points)
Compliance	Compliance with financial industry regulations.	3 - Excellent
Support and Maintenance	Quality of customer support and regular maintenance updates.	2 - Good
Scalability	How well the vendor's solutions can adapt to growth.	2 - Good
Compatibility	Compatibility with existing or planned system architecture.	2 - Good

Cloud Services Recommended Vendors

Vendor Name	Vendor Strengths	Vendor Weaknesses	Products/Services Provided to Project	Aggregate Selection Criteria Score
Amazon Web Services (Amazon Web Services, Inc., 2023)	Excellent reputation, strong compliance record.	Potential for high cost at scale.	AWS cloud computing services	9
Microsoft (Microsoft, 2023)	Good reputation, extensive suite of integrated tools.	Less straightforward pricing model.	Microsoft Azure services	8

Cloud Services Deployment Challenges

Deployment Challenge	Deployment Challenge Description

Cost Management	Keeping track of multiple services and their costs.
Compliance	Meeting financial industry regulations and standards.
Risk Mitigation	Handling issues where a cloud service doesn't integrate well during initial setup

Technology Adoption Methods

Method Name	Method Description
Phased Transition	Moving services to the cloud in phases.
Pilot Test	Small-scale test with some users before full implementation.
Backup Plan	Contingency plan for unexpected challenges during the transition

Cost/Benefit Considerations

Benefits	Costs	Considerations
Operational Flexibility	Data Transfer Costs, Ongoing Maintenance and Operation Costs	Vendor Lock-in
Geographic Reach: Data centers closer to users	Reserved Instances: Long-term commitment costs	Security: Meeting industry-specific security standards

References

Amazon Web Services, Inc. (2023). *Cloud Computing Services*. Amazon Web Services (AWS).

https://aws.amazon.com/

Cisco Systems, Inc. (2023). *Cisco Financial Services Solutions*. Cisco.

https://www.cisco.com/c/en/us/solutions/industries/financial-services.html

Finastra. (2023). *Fusion Essence | Core Banking Software*. Finastra.

https://www.finastra.com/solutions/fusion-essence

McAllister, C. (2023, July 20). *The limitations of PostgreSQL in financial services*. Cockroach Labs.

https://www.cockroachlabs.com/blog/limitations-of-postgres/

Microsoft. (2023). *Azure for Financial Services*. Microsoft Azure.

https://azure.microsoft.com/en-us/solutions/industries/financial

MongoDB, Inc. (2023). *Better Bank Experiences At A Lower Cost With MongoDB*. MongoDB.

https://www.mongodb.com/industries/financial-services

Oracle. (2023). *Discover the benefits of the Oracle fintech ecosystem.* Oracle.

https://www.oracle.com/financial-services/fintech/

Palo Alto Networks. (2023). *Financial Services*. Palo Alto Networks.

https://www.paloaltonetworks.com/industry/financial-services

Temenos Headquarters SA. (2023). *Temenos Transact – Core Banking Software (T24 is now Transact)*.

Temenos. https://www.temenos.com/products/transact/

Supporting Research Report

Fill out each section with information relevant to your project. Be sure to include the name and purpose of your project.

Supporting Research Report for: CAPS LOCK Bank

Project name: Internet Banking Digital Platform Implementation

Purpose of project: The project's primary goals are to launch a comprehensive digital banking platform that simplifies financial management and enhances user experience, aligning with modern banking demands and customer needs. The platform should be user-friendly, secure, and scalable, reflecting the business context and delivering the specified features and functions.

Executive Summary

CAPS LOCK Bank is launching an Internet Banking Digital Platform to offer comprehensive digital financial services in response to the growing preference for digital banking. With a focus on user-friendliness, security, and scalability, our platform aims to simplify financial management while enhancing the user experience. Utilizing Agile methodology and a hybrid development strategy, the project blends in-house development with third-party solutions to foster compliance and efficiency in delivering a customer-centric platform.

Industry Background

The digital banking sector is characterized by a strong trend towards digital-only services, with an uptick in consumer preference for digital banking due to unparalleled convenience, accessibility, and a tailored user experience. (Underwood & Aldrich, 2023) Concurrently, there's a notable emergence of FinTech startups offering innovative solutions that are reshaping the traditional banking paradigm. On the regulatory front, digital banking platforms, although innovative, must adhere to stringent regulations, compliance standards, data protection mandates, and customer rights.

Technology Trends

According to both Stratoflow and The Financial Brand, emerging technologies in the digital banking landscape include advanced Digital Banking Platforms, AI, Cloud Computing, Open Banking

APIs, and stringent Cybersecurity measures. AI enhances personalized banking and improves operational efficiency, while cloud computing supports the expansive data and processing needs of digital banks, providing necessary scalability and flexibility. Open Banking APIs are crucial for the seamless integration of services and products, enhancing the consumer experience. (Marous, 2021) Additionally, with the rise of digital banking, robust cybersecurity has become imperative to protect sensitive data and ensure secure transactions.

Project Approach

The adoption of the Agile methodology was and still is our first choice for approach, courtesy of its flexibility and efficiency in managing IT projects. Agile allows for continual adjustments and refinements to the project as it progresses, with an emphasis on customer feedback and iterative testing to maintain a customer-centric focus. A hybrid development strategy is going to be the path we take. Integrating in-house development with external third-party solutions for specialized services. This way we can ensure compliance and regulations are met when it is necessary without having to do too much work in house if there are already proper solutions on the market.

Alternative Approach

An alternative use of the waterfall method could be considered; but as fully digital banking is not as established as traditional banking systems and is evolving at a much faster pace, we don't think this approach would work out too well with quickly changing requirements. Another consideration could be building the entire system in-house without using any sort of specialized solutions. This would give us full control over all parts of the project but also could require much more time and financial commitment to get to the fully shipped product compared to the hybrid approach that we intend to take.

Impact Analysis

The launch of the platform is expected to positively influence business metrics, with an immediate focus on customer acquisition and retention. By offering a comprehensive suite of digital financial services through a user-centric platform, the bank anticipates fostering customer loyalty and satisfaction. The

intuitive interface, streamlined transaction process, and personalized services are designed to enhance the user experience significantly, making financial management more accessible and efficient for all users.

Risk Analysis

The platform faces various risks including technical glitches, security breaches, and operational challenges. Technical risks range from minor user experience issues to major functional challenges. Proactive identification and resolution of these issues are crucial, especially as we roll out a completely new system. Security risks are addressed through implementing industry-leading practices such as multi-level authentication and encryption to safeguard user data. To reduce operational risks like compliance issues, human errors, and system downtimes, ongoing staff training and following procedural guidelines are crucial. (*FinTech Compliance in 2023: Everything You Need To Know*, n.d.)

References

FinTech Compliance in 2023: Everything You Need To Know. (n.d.). SpdLoad. https://spdload.com/blog/fintech-compliance/

Marous, J. (2021, November 22). *Six Retail Banking Technology Trends for 2022*. The Financial Brand. https://thefinancialbrand.com/news/banking-trends-strategies/2022-retail-banking-technology-trends-ai-api-cloud-rpa-cybersecurity-125542/

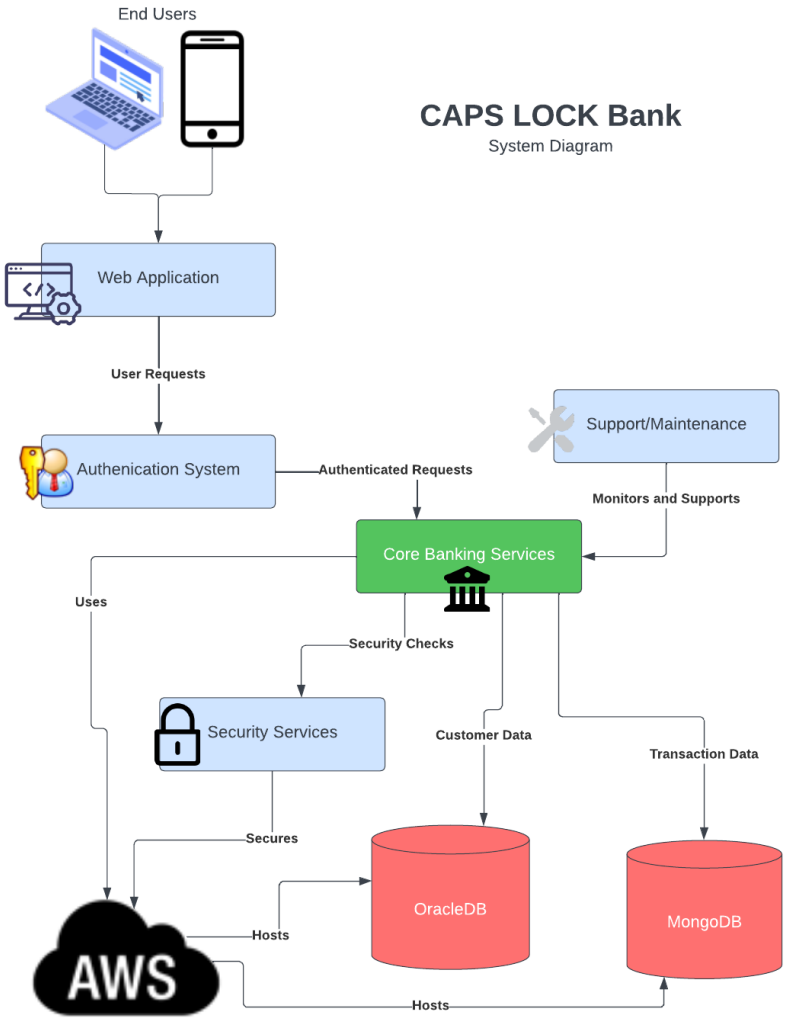
Pajorska, Z. (2023, April 19). *Banking Technology: Top 7 Trends for 2023*. Stratoflow. https://stratoflow.com/banking-technology-trends/

Underwood, J., & Aldrich, E. (2023, March 24). *Consumer Banking Trends and Statistics 2023 – Forbes Advisor*. Forbes. https://www.forbes.com/advisor/banking/banking-trends-and-statistics/

Project Documentation

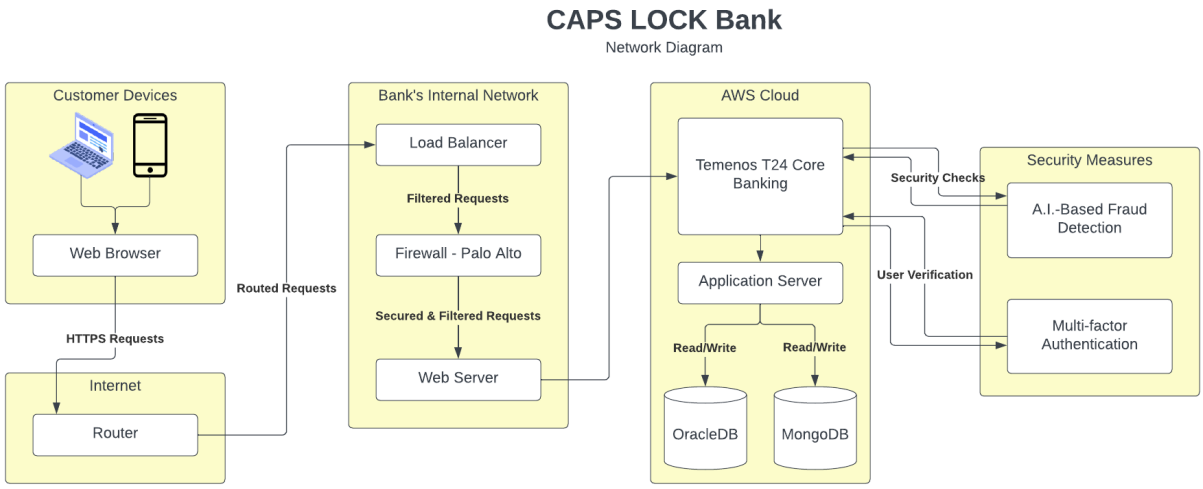
System Diagram

The Systems Diagram provides a high-level overview of CAPS LOCK Bank's Digital Banking Platform architecture. The diagram visually delineates the primary system components, their functions, and how they interact to offer seamless banking services to the users.



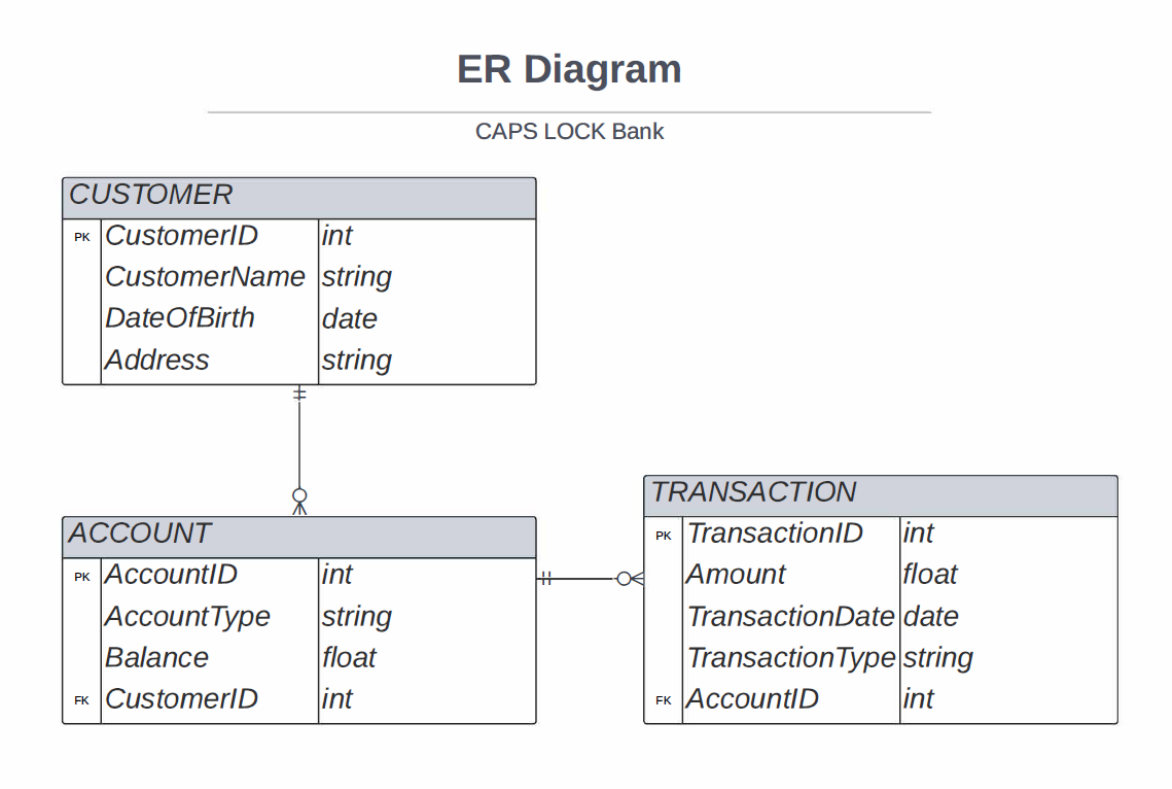
Network Diagram

Our Network Diagram maps out the data flow and connectivity within the platform. It graphically represents the network structure, displaying various network nodes and the connections between them. This depiction provides an understanding of how data is transmitted and received within the system.



Database Design

The DB Design documentation serves as a blueprint for our database structure, outlining the way data is organized and interrelated within the platform. It includes an entity-relationship diagram, showcasing the relationships between different data entities, a data dictionary providing details on data types and constraints, and table definitions that further elaborate on the purpose and structure of each table within the database.



Data Dictionary

Table: CUSTOMER

- **CustomerID:**
  - Type: Integer
  - Description: Unique identifier for each customer.
  - Constraints: Primary Key, Not Null, Auto-increment.
- **CustomerName:**
  - Type: VARCHAR(255)
  - Description: Full name of the customer.
  - Constraints: Not Null.
- **DateOfBirth:**
  - Type: DATE
  - Description: Customer's date of birth.
  - Constraints: Not Null.
- **Address:**
  - Type: VARCHAR(255)
  - Description: Customer's home address.
  - Constraints: Not Null.

Table: ACCOUNT

- **AccountID:**
  - Type: Integer
  - Description: Unique identifier for each account.
  - Constraints: Primary Key, Not Null, Auto-increment.
- **CustomerID:**
  - Type: Integer
  - Description: Identifier linking to the customer who owns the account.
  - Constraints: Foreign Key referencing CUSTOMER(CustomerID), Not Null.
- **AccountType:**
  - Type: VARCHAR(50)
  - Description: Type of account (e.g., Checking, Savings).
  - Constraints: Not Null.
- **Balance:**
  - Type: DECIMAL(15,2)
  - Description: Current balance of the account.

- Constraints: Not Null.

Table: TRANSACTION

- **TransactionID:**
  - Type: Integer
  - Description: Unique identifier for each transaction.
  - Constraints: Primary Key, Not Null, Auto-increment.
- **AccountID:**
  - Type: Integer
  - Description: Identifier linking to the account associated with the transaction.
  - Constraints: Foreign Key referencing ACCOUNT(AccountID), Not Null.
- **Amount:**
  - Type: DECIMAL(15,2)
  - Description: Amount of money involved in the transaction.
  - Constraints: Not Null.
- **TransactionDate:**
  - Type: DATETIME
  - Description: Date and time when the transaction occurred.
  - Constraints: Not Null.
- **TransactionType:**
  - Type: VARCHAR(50)
  - Description: Type of transaction (e.g., Deposit, Withdrawal).
  - Constraints: Not Null.

Table Definitions

CUSTOMER Table

- Stores information about the bank's customers.
- Relationships:
  - One-to-Many relationship with the ACCOUNT table through CustomerID.

ACCOUNT Table

- Stores information about customer accounts.
- Attributes include AccountID, CustomerID, AccountType, and Balance.
- Relationships:
  - Many-to-One relationship with the CUSTOMER table through CustomerID.

    - One-to-Many relationship with the TRANSACTION table through AccountID.

TRANSACTION Table

- Stores information about transactions occurring in customer accounts.
- Attributes include TransactionID, AccountID, Amount, TransactionDate, and TransactionType.
- Relationships:
    - Many-to-One relationship with the ACCOUNT table through AccountID.

Cybersecurity Plan

CAPS LOCK Bank prioritizes customer data protection and platform integrity. This plan presents strategies for safeguarding data and processes against unauthorized interventions. We are dedicated to securing customer data through robust measures. With a commitment to regular plan reviews and updates in response to the evolving cybersecurity environment.

Framework:

- **Zero Trust:** Enforcing strict identity verification for every user and device.
- **Multi-Factor Authentication (MFA):** Enhancing access controls with multiple verification methods.
- **Software-Defined Networking (SDN):** Centralizing network management and optimizing traffic routing.

Data Protection:

- **Encryption:** Employing advanced technologies for data protection.
- **Database Security:** Applying access controls, monitoring, and policies to secure customer data.

Network Security:

- **Firewalls:** Utilizing Next-Gen Firewalls for network security.
- **Secure Access:** Implementing secure and efficient network access with Cisco's SDN controllers and switches.
- **Intrusion Detection:** Monitoring network activities and policy violations.

Application Security:

- **Secure Coding**: Following best practices to prevent application-layer attacks.
- **Security Audits:** Performing regular assessments to identify and mitigate vulnerabilities.

Response Plan:

- Establishing a plan for effective incident identification and response.

Compliance:

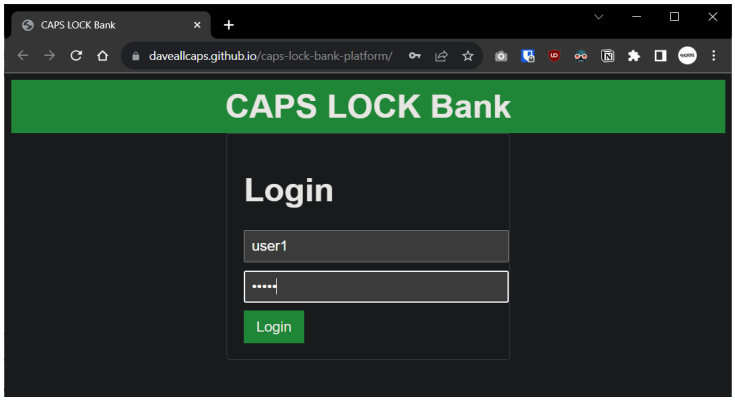
- Maintaining compliance with relevant industry regulations and standards.

Source Code

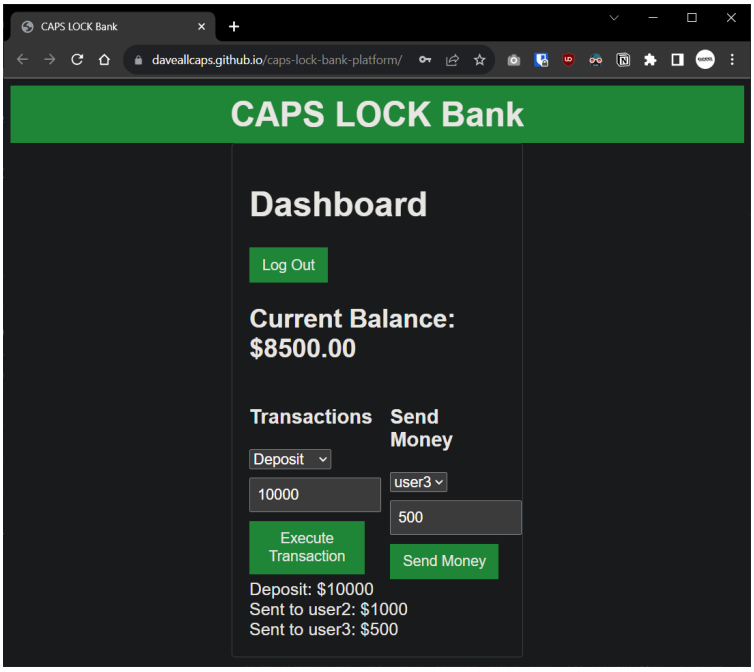
The project comprises a simple, front-end only, digital banking application developed using HTML, CSS, and JavaScript. The application supports functionalities such as login/logout, balance view, deposit/withdrawal, and fund transfers between predefined users. With three predefined user accounts to transact between, updating the balance and transaction history in real time, the application provides a straightforward demonstration of a banking application interface. I have the full code hosted on GitHub github.com/DAVEALLCAPS/caps-lock-bank-platform

Live version of the application is available at daveallcaps.github.io/caps-lock-bank-platform/

Login Page Screenshot



Account Dashboard Screenshot



Now that we have taken a look at the mock banking web app, here is the code that makes it work.

index.html file

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CAPS LOCK Bank</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <h1 class="header">CAPS LOCK Bank</h1>
    <div id="login-container">
      <h1>Login</h1>
      <input type="text" id="username" placeholder="Username" />
      <input type="password" id="password" placeholder="Password" />
      <button onclick="login()">Login</button>
    </div>
    <div id="dashboard" style="display: none">
      <h1>Dashboard</h1>
      <button onclick="logout()">Log Out</button>
      <h2>Current Balance: $<span id="balance">0</span></h2>
      <div
        id="forms-container"
        style="display: flex; justify-content: space-between"
      >
        <div id="transaction-form" style="width: 45%; display: none">
          <h3>Transactions</h3>
          <select id="transaction-type">
            <option value="Deposit">Deposit</option>
            <option value="Withdraw">Withdraw</option>
          </select>
          <input type="number" id="transaction-amount" placeholder="Amount" />
          <button onclick="addTransaction()">Execute Transaction</button>
        </div>
        <div id="send-money-form" style="width: 45%; display: none">
          <h3>Send Money</h3>
          <select id="recipient"></select>
          <input type="number" id="send-amount" placeholder="Amount" />
          <button onclick="sendMoney()">Send Money</button
        </div>
      </div>
      <div id="transactions" style="display: none"></div>
    </div>
    <script src="script.js"></script>
  </body>
</html>
```

https://github.com/DAVEALLCAPS/caps-lock-bank-platform/blob/main/index.html

script.js file

```
const accounts=[{username:"user1",password:"pass1",balance:0,transactions:[]},
{username:"user2",password:"pass2",balance:0,transactions:[]},
{username:"user3",password:"pass3",balance:0,transactions:[]},];

let currentAccount = null;

function login() {
  const username = document.getElementById("username").value;
  const password = document.getElementById("password").value;

  currentAccount = accounts.find(
    (acc) => acc.username === username && acc.password === password
  );

  if (currentAccount) {
    document.getElementById("login-container").style.display = "none";
    document.getElementById("dashboard").style.display = "block";
    document.getElementById("transaction-form").style.display = "block";
    document.getElementById("send-money-form").style.display = "block";
    recalculateBalance(currentAccount);
    updateBalanceDisplay();
    populateRecipients();
    displayTransactions();
  } else {
    alert("Invalid credentials");
  }

  recalculateBalance(currentAccount);
  updateBalance();
  populateRecipients();
  document.getElementById("send-money-form").style.display = "block";
}

function populateRecipients() {
  const recipientSelect = document.getElementById("recipient");
  recipientSelect.innerHTML = "";
  accounts.forEach((acc) => {
    if (acc !== currentAccount) {
      const option = document.createElement("option");
      option.value = acc.username;
      option.textContent = acc.username;
      recipientSelect.appendChild(option);
    }
  });
}

function sendMoney() {
  const recipientUsername = document.getElementById("recipient").value;
  const amount = parseFloat(document.getElementById("send-amount").value);
  if (isNaN(amount) || amount <= 0) {
    alert("Please enter a valid amount.");
    return;
  }
  if (amount > currentAccount.balance) {
    alert("Insufficient funds.");
    return;
  }

  const recipientAccount = accounts.find(
    (acc) => acc.username === recipientUsername
  );
```

```
  currentAccount.transactions.push({
    type: `Sent to ${recipientUsername}`,
    amount,
  });
  recipientAccount.transactions.push({
    type: `Received from ${currentAccount.username}`,
    amount,
  });

  recalculateBalance(currentAccount);
  recalculateBalance(recipientAccount);
  updateBalanceDisplay();
  displayTransactions();
}

function recalculateBalance(account) {
  account.balance = 0;
  account.transactions.forEach((transaction) => {
    if (
      transaction.type.startsWith("Received") ||
      transaction.type === "Deposit"
    ) {
      account.balance += transaction.amount;
    } else {
      account.balance -= transaction.amount;
    }
  });
}

function displayTransactions() {
  const transactionsContainer = document.getElementById("transactions");
  transactionsContainer.innerHTML = "";
  currentAccount.transactions.forEach((txn) => {
    const txnDiv = document.createElement("div");
    txnDiv.textContent = `${txn.type}: $${txn.amount}`;
    transactionsContainer.appendChild(txnDiv);
  });
  transactionsContainer.style.display = "block";
}

function addTransaction() {
  const type = document.getElementById("transaction-type").value;
  const amount = parseFloat(
    document.getElementById("transaction-amount").value
  );
  if (isNaN(amount) || amount <= 0) {
    alert("Pease enter a valid amount.");
    return;
  }
  currentAccount.transactions.push({ type, amount });
  recalculateBalance(currentAccount);
  updateBalanceDisplay();
  displayTransactions();
}

function updateBalanceDisplay() {
  document.getElementById("balance").textContent =
    currentAccount.balance.toFixed(2);
}

function logout() {
  document.getElementById("login-container").style.display = "block";
  document.getElementById("dashboard").style.display = "none";
  document.getElementById("transaction-form").style.display = "none";
  document.getElementById("transactions").style.display = "none";
  document.getElementById("username").value = "";
  document.getElementById("password").value = "";
  currentAccount = null;
}
```

https://github.com/DAVEALLCAPS/caps-lock-bank-platform/blob/main/script.js

Software Test Plan

Project name: CAPS LOCK Bank Platform

Purpose of project: To demonstrate a simple digital banking platform with fundamental features such as login, balance checking, transactions (deposit and withdrawal), and fund transfers.

Features To Be Tested/Not To Be Tested

- **To be Tested:**
    - Login with predefined users.
    - Balance checking and display.
    - Deposit and Withdraw transactions.
    - Fund transfer between users.
    - Transaction history display.
- **Not to be Tested:**
    - Real-time database integration (since it's a front-end only application).
    - Security features, such as encryption and secure connection.

Testing Pass/Fail Criteria

Each feature should work as expected without causing any errors or crashing the application.

Testing Approach

- **Manual Testing:** The application will be tested manually by the testing team using different browsers to ensure compatibility and responsiveness.
- **Functional Testing:** Ensures that the functional features work as expected.
- **Usability Testing:** To ensure the application is user-friendly and meets the end user's expectations.

Testing Cases

1. Test login with each predefined user.
2. Test login with incorrect credentials.

3. Check initial balance for each user.
4. Test deposit function for each user.
5. Test withdrawal function (valid and overdraw scenarios).
6. Test fund transfer function (valid and insufficient funds scenarios).
7. Validate transaction history updates after each transaction.
8. Validate balance updates after each transaction.
9. Check the UI responsiveness and compatibility with different browsers.
10. Logout functionality test.
11. Test navigation between different sections of the application.
12. Validate error messages and alerts are displayed correctly.

Testing Materials (Hardware/Software Requirements)

- Computer with internet access
- Web browsers: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge
- No additional software or hardware required.

Testing Schedule

Testing Activity	Duration	Resource	Comments
Test Plan Creation	2 days	Test Manager	Outline of testing objectives, scope, and methods.
Test Specification Creation	3 days	Test Leads	Detailed documentation of test scenarios and expected outcomes.
Test Specification Team Review	1 day	Project Team	Review to ensure test cases cover all requirements and are feasible.

Testing Activity	Duration	Resource	Comments
Manual Testing	2 days	Test Leads	Manual execution of test cases to validate functionality.
Functional Testing	2 days	Component Testers	Testing of individual functionalities to ensure they work as intended.
Usability Testing	2 days	System Testers	Evaluating the system's user interface and overall user experience.
Compatibility Testing	2 days	System Testers	Ensuring the system works across different devices, browsers, and OS.
Result Analysis and Report	2 days	Test Manager	Compilation and review of test results, identification of defects, and recommendations.
	Total: 16 days		

Risks and Contingencies Matrix

Risk	Probability	Risk Type	Owner	Contingencies/Mitigation Approach
Unidentified bugs/issues during testing.	50%	Quality Risk	Test Leads	Implement thorough test cases; continuous testing to identify and fix bugs early.

Risk	Probability	Risk Type	Owner	Contingencies/Mitigation Approach
Inaccurate test cases leading to flawed testing.	20%	Quality Risk	Test Leads	Regularly review and update test cases to ensure accuracy and relevance.