Requirement Analysis Document.

**1 Introduction**

* 1. Purpose of the system

The purpose of this project is to create a database system that has multiple replicated databases that are stored in different servers with a major back up database for synchronization. The database system is to be implemented to the bank ATM system. A single database system controls all ATM transactions for their multiple ATM. The project aims at using database partitioning and replication to create a system that will simplify the management of the ATM bank transactions. The system will be an upgrade of the current system which provides several challenges in its functionality and maintainability. The system’s main purpose is to improve scalability which in turn improves availability in addition to increasing the manageability of the database. Upgrades and configurations to the database can be made without interfering with the normal functionality of the system.

* 1. Scope of the system

The scope of the system is the bank account database. The system will only be used in managing the ATM transaction processes. The system will apply in the access of the account database. In the implementation of the system, it will run through all the partitioned databases and their consequent servers. It will also apply to the backup database (main database) and its server. The system is applicable to one bank at a time and all its ATMs.

* 1. Objectives and success criteria of the project

The main objective of the system is to overcome the challenges brought about by using a single server to run all ATM transaction processes. This will be accomplished through the following minor objectives;

* Replication of the database
* Partitioning of the database
* Configure the partitions to synchronize and match the main database once any change is made to any database
* Configure access to the database to hold one transaction at a time and lock out other transaction when there is one entry
* Faster ATM transaction processing
* Synchronized data in all databases at all times
* Successful distribution of transactions
* Improved general performance
* Finer levels of granularity
  1. Definition, acronyms and abbreviations
* Database replication – process of copying and maintaining the data objects in multiple databases
* Database partition – this is the dividing of the database tables into multiple tables. Vertical partitioning creates multiple tables with fewer columns same rows while horizontal partitioning creates multiple tables with fewer rows but same number of columns.
* Database Synchronization – this refers to the keeping of a number of databases, two or more, separate from each other up to date with the changes of each other.
  1. References

Refers to the current database system used by the banks

* 1. Overview

The successful implementation of the system can be adopted by a variety of bank and other applications that require distributed database management. The systems main advantage is increased availability and speed of access.

1. **Current system**

Banks operate several ATMs at any given time. This is achieved through one single database in the current system. The single database operates from one server as well and there is no presence of a backup database or server in most cases. This system scenario comes with a number of challenges that the new system seeks to solve. The challenges identified in the system include the risk associated with loss of data in case the server fails or corrupt database. There is also the challenge associated with low speed in accessing the database from a variety of ATMs. Reliability and availability is a challenge with the current system. Reliability issues are also present when the database or server requires updating or configuring.

1. **Proposed System**
   1. Overview

The proposed system is a distributed database system. The system is to implement replicated and partitioned databases of the main database. The main database will have a backup database with its own server dedicated to it. The replicated and partitioned databases will have each their own servers. The ATM transaction will utilize any available database during access. If the process in one database will be too much for it to handle, the process will be distributed among other various databases. The databases will be connected in such a way that if one of the servers or databases was to stop working, there would be no delay to any of the ATM transactions from any ATM. The system aims at increasing the speed of the transactions as well.

**3.2 Functional Requirements**

* **Databasesynchronization**
* `When any changes are made to the main database, the backup database will be simultaneously updated. The purpose of this is so both databases has accurate and dependable information.
* **One transaction at a time for each database**
* For security purposes, the database will lock out any attempted transaction from entry if there is already a transaction in progress. Once a transaction begins, the database is committed to completing that task. The attempted transaction will gain access once the current transaction is complete.
* **Database partitioning**
* Databases that exceed the 2 gigabyte limit will be partitioned. Partitioning is when you divide database table into multiple tables. This is effective because accessing the information would now be extremely faster and easier.
* **Database replication**
* When the ATM machine sends a request to the database a specific number of times, the server system would time out. When that happens, the database will send that same request to another server. The requests will then be distributed among the database servers with all of the modifications included.

3.3 Non-functional Requirements

3.3.1 Usability

This is the user experience when using the system. The user will not need to learn any new utilization of the system since only the background running will change. Handling of

errors will not affect the user functionality.

3.3.2 Reliability

This will be achieved through the replication of databases once the requests exceed the

specified level. It will increase the availability brought about by the systems scalability.

This will also allow for updating of the systems without interfering with the customers

interactions.

3.3.3 Performance

The speed of ATM transaction processes will be increased through the partitioning of the

database tables.

3.3.4 Supportability

The system will be scalable and can grow without interfering with all the other databases.

The system will be easy to maintain considering the several databases when one server is

under maintenance, transactions will proceed as required by the system and the customer.

3.3.5 Implementation

The system will be implemented alongside the current system and any challenges

witnessed changes before the system runs alone. The coding and platform will be similar

to the current system therefor there will be few changes required.

3.3.6 Interface

To reduce the amount of learning required by the users, the system will be applied in

such a way that the interaction with the external system will be similar to that of the

current system. The protocols will also be changed where necessary as well as the nature

of passing information across the interface.

3.3.7 Packaging

The system will be installed by the developers in the presence of the bank IT manager.

There will be only one installation where the system will run alongside the current system

and any changes will be made when the system is still running until they are finalized.

There are no time constraints anticipated.

3.3.8 Legal

The system will be licensed in the banks name. Once the system runs efficiently during

the first installation, there is not overall failure anticipated.

3.4 System Models

3.4.1 Scenarios

|  |  |
| --- | --- |
| Scenario | Account access |
| Participants | Account owner and system |
| Flow of events | 1. Insert the card in the ATM machine 2. ATM machine requests for pin 3. System checks if the pin is correct 4. System asks which account customer wants to access; savings, checking or loan 5. The customer selects the account to access 6. The system asks the customer which transaction to make; withdrawal or check balance 7. The system also requests to provide a receipt alongside the chosen transaction 8. The customer agrees or rejects the system request 9. The system process the transaction 10. The system provides the customer with the request needs 11. The system asks the customer if there is another transaction 12. The system proceeds as answered 13. When the customer answer no, the system asks the customer to collect the card 14. The system provides the receipt if requested 15. The system displays a thank you message and logs to the start screen |

3.4.2 Use case model

|  |  |
| --- | --- |
| Use case name | Account withdrawal |
| Participating actors | Customer and system |
| Flow of events | 1. Customer insert’s the card in the ATM machine 2. ATM machine requests for pin 3. System checks if the pin is correct 4. System asks which account customer wants to access; savings, checking or loan 5. The customer selects the account to access 6. The system asks the customer which transaction to make; withdrawal or check balance 7. Customer chooses the withdraw transaction 8. System request for the amount to withdraw 9. Customer enters the amount to withdraw 10. System asks customer if they want a receipt 11. Customer choses yes or no 12. System process the request 13. System provides the amount of money requested 14. If customer requires no other transaction the system ejects the card and provides receipt if requested to do so 15. System thanks the customer and exits to the start page |
| Entry conditions | Customer has an account  Customer has a card and the entered pin is correct |
| Exit conditions | The pin provided is wrong  The account has no sufficient amount of money  Customer prompts the exit by pressing cancel at any stem of the transaction |

|  |  |
| --- | --- |
| Use case name | Account balance |
| Participating actors | Customer and system |
| Flow of events | 1. Customer insert’s the card in the ATM machine 2. ATM machine requests for pin 3. System checks if the pin is correct 4. System asks which account customer wants to access; savings, checking or loan 5. The customer selects the account to access 6. The system asks the customer which transaction to make; withdrawal or check balance 7. Customer chooses the check balance transaction 8. System asks customer if they want a receipt 9. Customer choses yes or no 10. System process the request 11. System provides the balance amount on the ATM screen 12. If customer requires no other transaction the system ejects the card and provides receipt if requested to do so 13. System thanks the customer and exits to the start page |
| Entry conditions | Customer has an account  Customer has a card and the entered pin is correct |
| Exit conditions | The pin provided is wrong  Customer prompts the exit by pressing cancel at any stem of the transaction |



3.4.3 Analysis object model

Object analysis of the current system with one database and many transactions from different customers at different ATM machines. A customer can access one account a time for one transaction from that account. Several customer can access the database through a number of transactions which the database is handling other requests.

The following analysis object model is to represent the new system with replicated and partitioned databases with the main database and the backup database. A customer can access one account at a time and one transaction from each account at a go. The difference is that the system can distribute the different transaction to different databases than update all the transactions to the main database. Each update to the main database is also processes in the backup database.



3.4.4 Dynamic model

Sequence diagram



* Activity diagram



3.4.5 User interface—navigational paths and screen mock-ups



4. **Glossary**

Main database – this is the database that collects all the information from all the replicated and partitioned databases. It could be a representation of the database in the current system.

Backup database – this is a replication of the main database as a whole only that it is accessed when the main database encounters any malfunction errors. It is updated each time the main database is updated.