|  |
| --- |
| **Final Documentation** |
|  |
|  |
| |  | | --- | | **Abhishek Kumar Ravi | abhix@greensyntax.co.in** | |

|  |
| --- |
| ATM |



Table Layout

* + Abstract
  + System Analysis
  + Feasibility Study
  + Requirement Specification
  + Snapshots
  + Dfds(level-0,level-1)
  + User Case Diagram
  + UML Diagram
  + Testing
  + Maintenance
  + Future Scope

Abstract

The **ATM System** is the project which is used to access their bank accounts in order to make cash withdrawals. Whenever the user need to make cash withdraws, they can enter their PIN number (personal identification number) and it will display the amount to be withdrawn in the form of 100’s 500’s and 1000’s. Once their withdrawn was successful, the amount will be debited in their account.

General Overview

The ATM will service one customer at a time. A customer will be required to enter ATM Card number, personal identification number (PIN) – both of which will be sent to the database for validation as part of each transaction. The customer will then be able to perform one or more transactions. Also customer must be able to make a balance inquiry of any account linked to the card.

System Analysis

**Need For the Software:**

Now a days every one very busy in their work.  So they feel that the job must be easier so the system is used to reduce their work which is done in the ATM system.  Instead of keeping lots of paper into a record or file and it may be missed somewhere so, this system help to keep the record of the customer it also keeps the details of he customer.  It is also easy to access.

**Problem description :**

The system mainly used by the bank clients.  When a client comes to ATM centre to update and delete their account.  It reduces the time consumption and lot of paperwork.  For any single operation it involves numerous references and updating also takes subsequent changes in other places.

Feasibility Study

It includes the following-

1. Technical Feasibility
2. Economical Feasibility
3. Operational Feasibility
4. **Technical Feasibility:**

Since the project is design with ASP.NET with C# as code behind and SQL server 2008 as backend, it is easy to install in the systems whenever needed. It is more efficient, easy and user friendly to understand by almost everyone. Huge amount of data can be handled efficiently using SQL server as backend. Hence this project has good technical feasibility. As per the technical aspect the project College Management System is technically suitable for the user as well as the developer.

1. **Economical Feasibility:**

Economic feasibility is mainly concern with the cost incurred during the implementation of the software. Since the project is developed using ASP.NET with C# and SQL server which is more commonly. It is financially feasible.  There is no need of spending over money.  Mainly this system constructed by existing devices only.  This project reduces the number of workers wage also.

1. **Operational Feasibility:**

As this ATM system is very easy to operate for the people and easy to use. So this project has a good operational feasibility.

Hence this project is technically, economically as well as operationally feasible.

­Requirement Specification

**Minimum Requirements for Client Side:**

**Hardware Requirement:**

* Intel i3 or above based PC
* 1 GB of RAM
* Minimum Space Required: 40GB.
* Display: 64 bit colour.

**Software Requirement:**

* Operating System                   – Microsoft Windows XP With Service Pack 2
* Front-End                                – Microsoft Visual Studio 20013
* DataBase                               – MS Access

**Performance Requirement:**

* The performance of the product is dependent on the hardware.

**Programming Language and Developing Tools:**

**Programming Language:**

* C#.

**Development Tools:**

* Visual Studio 2013
* MS-Access
* Operating System:- Windows 7

Directory Module

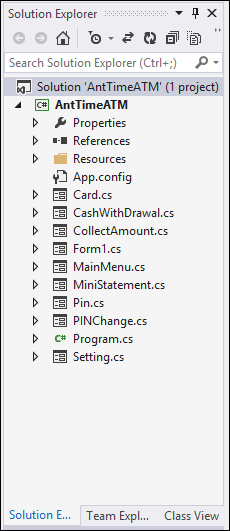


Fig1: Insert YOUR ATM .

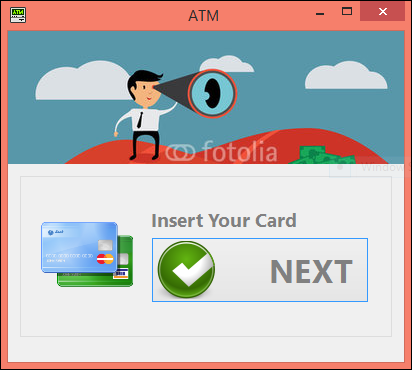


Fig2: YOUR A/C NO:

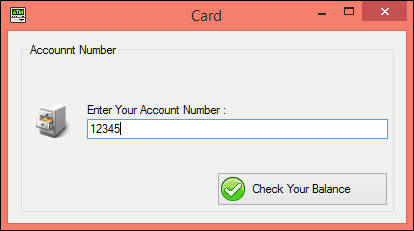


Fig3: Now Fill The PIN Section

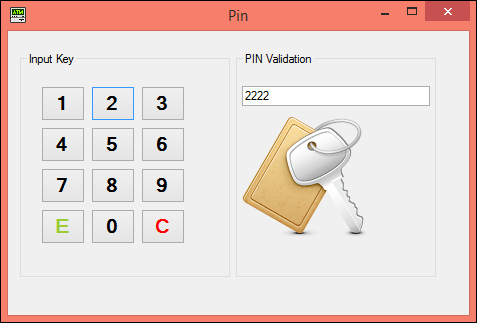


Fig4: Select the operation YOU want:

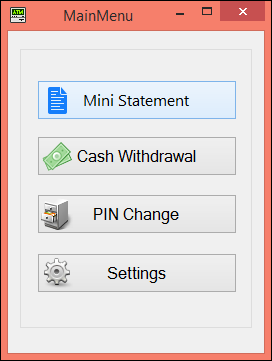


Fig5: For Ministatement:

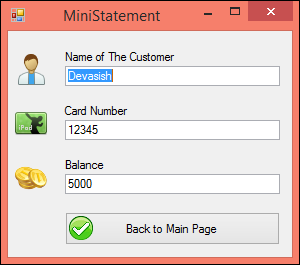


FIG -6: CashWithdraw:

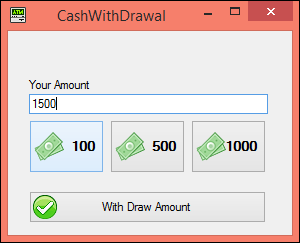


FIG-7: Collect Amount

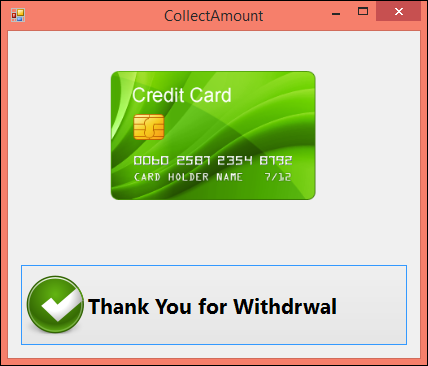


FIG-8: For PIN change:

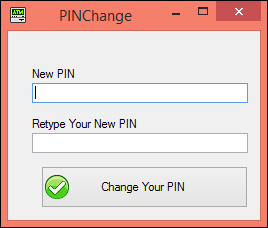
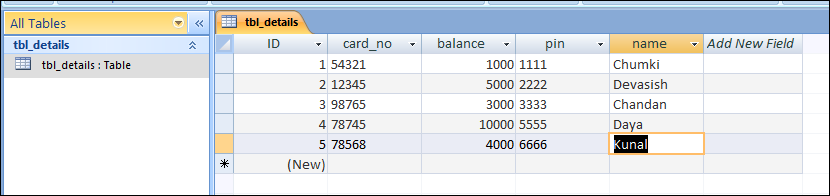
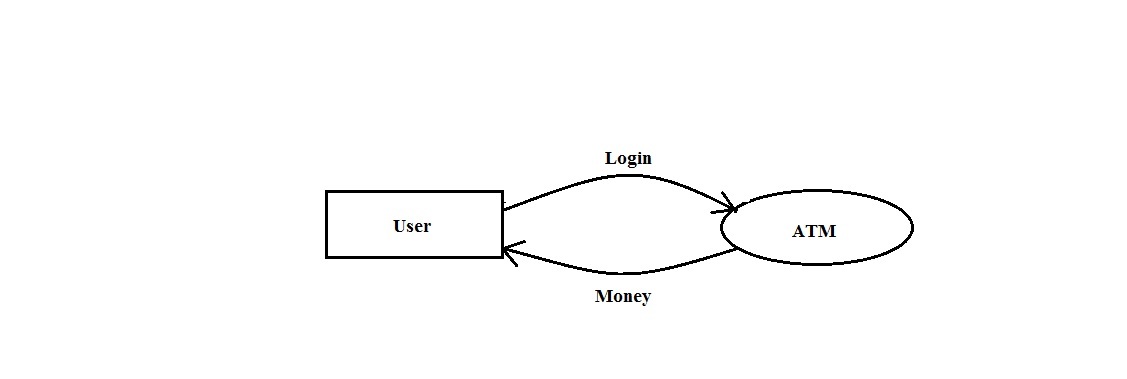


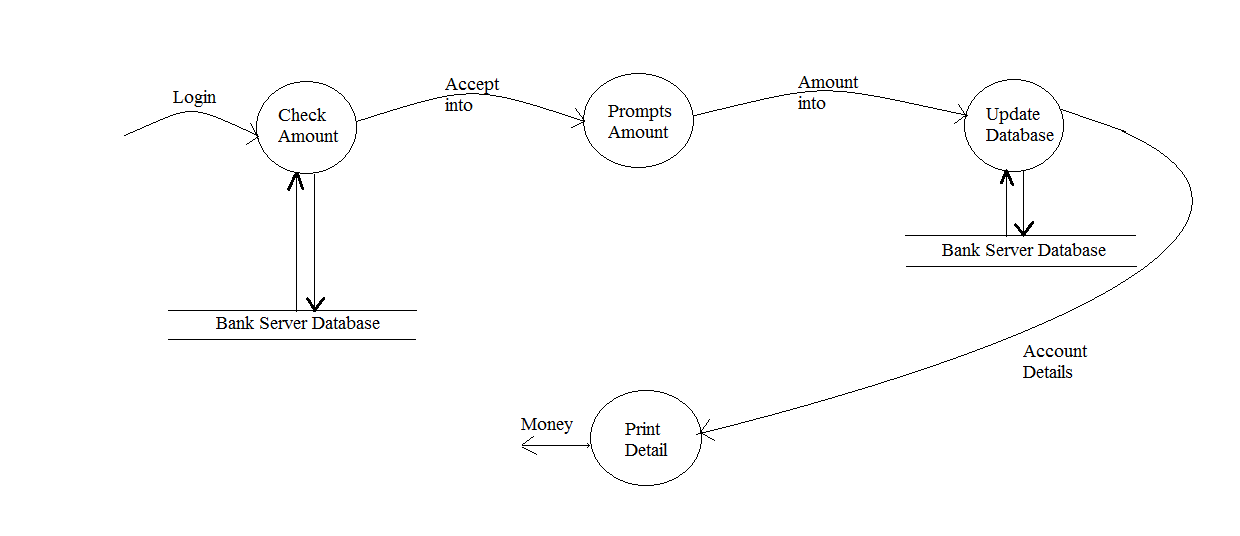
FIG-9: DATABASE:



DFDs:



DFD level 0



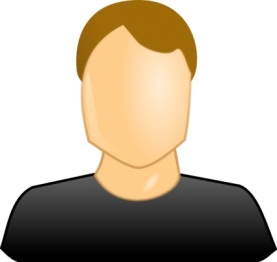
DFD level -1

User Case Diagram

Payment Utility

****

**Change PIN**



**Get Statement**

User

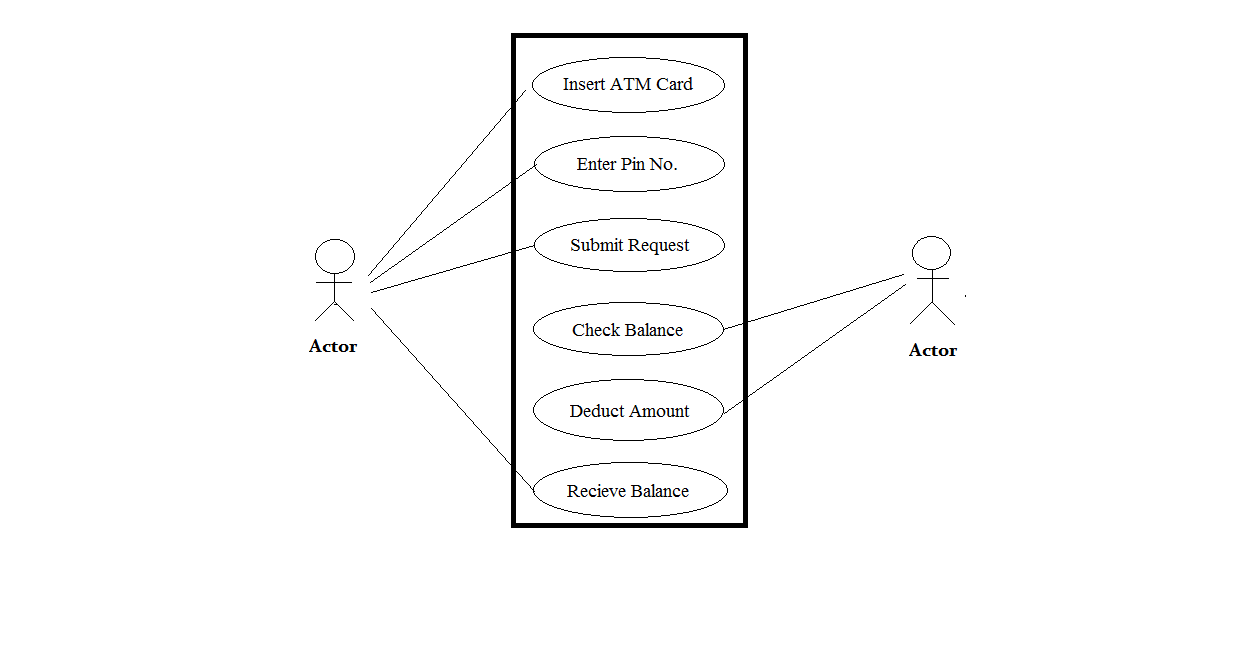
**Activate ATM**



**Deactivate ATM**

Administrator

UML Diagram:



Testing

* **White Box Testing:**

**White-box testing** (also known as **clear box testing**, **glass box testing,transparentbox testing** and **structural testing**) is a method of testing [software](http://en.wikipedia.org/wiki/Software) that tests internal structures orworkings of an application.

Using **white**-**box** **testing** methods, we derived test cases that:

* Guarantee that all independent paths within a module have been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds, and
* Exercise internal data structures to ensure their validity.

The goal of white box testing is to cover testing as many of the statements, decision point, and branches in the code base as possible.

# **Black Box Testing:**

Black-box testing, also called **behavioral**testing, focuses on the method of [software testing](http://en.wikipedia.org/wiki/Software_testing) that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see [white-box testing](http://en.wikipedia.org/wiki/White-box_testing)).

**Black**-**box** **testing** attempts to find errors in the following categories:

* incorrect or missing functions,
* interface errors,
* errors in data structures or external database access,
* behavior or performance errors, and
* Initialization and termination errors.
* **Unit Testing:**

**Unit testing** refers to tests that verify the functionality of a specific section of code, usually at the function level.

These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

* **Integration Testing:**

**Integration testing** (sometimes called Integration and Testing, abbreviated "I&T") is the phase in software in which individual software modules are combined and tested as a group. It occurs after [unit testing](http://en.wikipedia.org/wiki/Unit_testing). Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items.

* **System Testing:**

**System testing** of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified [requirements](http://en.wikipedia.org/wiki/Requirements). System testing takes, as its input, all of the "integrated" software components that have successfully passed [integration testing](http://en.wikipedia.org/wiki/Integration_testing) and also the software system itself integrated with any applicable hardware system(s).

Maintenance

Maintenance is the enigma of system development. Maintenance can be classified as corrective, adaptive and perfective.

Adaptive maintenance means changing the program function.

Perfective maintenance means enhancing the performance or modifying the program to respond to the user additional or changing needs.

Maintenance covers a wide range of activities, including correcting coding and design errors, updating documentation and test data and upgrading user support. Several MIS organization has done to attack the ever growing problem of software maintenance through a maintenance reduction-plan that consists of two phases.

i).Maintenance management audit.

ii).Software system audit

Maintenance management audit is done through interviews and questionnaires, evaluates the quality of maintenance effort.

Software system audit entails an overall view of the system documentation and an assessment of the quality of data files and database and system reliability and efficiency.

Software implementation which consists of program writes system level update and re audit of manual system to make sure that the errors have been corrected.

Future Scope

* Implement the Graphical Representation of the Data/collection.
* Re-design the Database architecture for better and cross-application support.
* Create more admin rights and privileges so that, they can customize the application according to themselves.
* Account transfer should be added in future.