**CHAPTER 1**

**INTRODUCTION**

In computing, a web application is a client–server software application in which the client (or user interface) runs in a web browser. Common web applications include webmail, online retail sales, online auctions, wikis, instant messaging services and many other functions.

In earlier computing models like client–server, the processing load for the application was shared between code on the server and code installed on each client locally. In other words, an application had its own pre-compiled client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server-side code of the application would typically also require an upgrade to the client-side code installed on each user workstation, adding to the support cost and decreasing productivity. In addition, both the client and server components of the application were usually tightly bound to a particular computer architecture and operating system and porting them to others was often prohibitively expensive for all but the largest applications.

In contrast, web applications use web documents written in a standard format such as HTML and JavaScript, which are supported by a variety of web browsers. Web applications can be considered as a specific variant of client–server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard procedures such as HTTP. Client web software updates may happen each time the web page is visited. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application.

In the early days of the Web each individual web page was delivered to the client as a static document, but the sequence of pages could still provide an interactive experience, as user input was returned through web form elements embedded in the page markup. However, every significant change to the web page required a round trip back to the server to refresh the entire page.

Later, the when client-side scripting language called JavaScript was introduced, it allowed programmers to add some dynamic elements to the user interface that ran on the client side. So instead of sending data to the server in order to generate an entire web page, the embedded scripts of the downloaded page can perform various tasks such as input validation or showing/hiding parts of the page.

In 1996, Macromedia introduced Flash, a vector animation player that could be added to browsers as a plug-in to embed animations on the web pages. It allowed the use of a scripting language to program interactions on the client side with no need to communicate with the server.

In 1999, the concept of "web application" was introduced in the Java language in the Servlet Specification version 2.2. At that time both JavaScript and XML had already been developed, but Ajax had still not yet been coined and the [XMLHttpRequest](https://en.wikipedia.org/wiki/XMLHttpRequest) object had only been recently introduced on Internet Explorer 5 as an ActiveX object.

In 2005, the term Ajax was coined, and applications like Gmail started to make their client sides more and more interactive. A web page script is able to contact the server for storing/retrieving data without downloading an entire web page.

In 2011, HTML5 was finalized, which provides graphic and multimedia capabilities without the need of client side plug-ins. HTML5 also enriched the semantic content of documents. The APIs and document object model (DOM) are no longer afterthoughts, but are fundamental parts of the HTML5 specification. [WebGL](https://en.wikipedia.org/wiki/WebGL) API paved the way for advanced 3D graphics based on HTML5 canvas and JavaScript language. These have significant importance in creating truly platform and browser independent [rich web applications](https://en.wikipedia.org/wiki/Rich_Internet_application).

The current data frameworks depend on the innovation called Database Management, which is an accumulation of consistent and related information. Database Management System fills in as a product framework which permits which permits the clients to characterize, make, keep up and control access to database. To execute this sort of frameworks is very little straightforward and simple. It includes a great part of the shared advancement of use program which is the part of an application engineer and database. The application program is, really the interface between the clients and the database, where the information is put away. Along these lines, an all-around created application program and database are essential for the unwavering quality, adaptability and usefulness of the framework.

The basic idea can be described on Figure below:

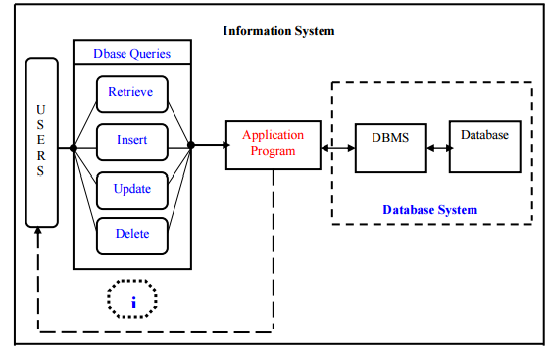


Figure 1.1 Information Flow in the System

Internet Banking is all about knowing the customer needs of a bank and provide them with the right service at the right time through right channel irrespective of the time and place of the user. Being “electronic”, it not only provides its customers with faster and better facilities, it even reduces the manual overhead of accounts maintenance.

The Online Banking suite provides a global accounting foundation that provides the all private banks with electronic banking facilities. It allows client of private banks to carry out their day to day banking transactions.

The developed system is an innovation in the area of private banking. In banking systems security is always a big question from the users perspective. And from the bank's perspective reduction of human resource and time involved in banking operations and avoiding errors caused due to manual mistakes is essential.

The data entry process requires the data on the paper, which is then feed into the application by the operator while doing so; the data entry operator has to look into the paper again & again and thus the chances of in accuracies in the typed contents increases. Also the process includes higher transportation cost, increased handling cost, more time delays, low accuracy, more usage of resources like registers, books, papers, etc Almost 60% of today’s information is still paper based. 30% of all office time is spent finding documents. The average time to manage a single document is 12 minutes, 9 minutes to re-file and 3 minutes to process. Hence the requirement is to develop a system that minimizes all these overheads included while giving the maximum output for the organization.

The basis for the project is to develop a fully automated banking system that includes depositing of amount, withdrawal of amount and exporting the outcome back to the client while considering all the tools and facilities than a client may need for efficient and effective output.

**Benefits of the system**

* Quick, authenticated access to accounts via internet.
* Easily scalable to grow with changing system requirement.
* Enterprise wide access to information.
* Improved information security, restricting unauthorized access.
* Minimize Storage Space

In manual system, much storage space for data files is required so to overcome this problem, on automated well managed database is developed for saving storage space. This s/w saves space and stores information efficiently. It ends the burden of having large manual filing storage system.

**Banking System can be used extensively**

* Withdrawal of amount by the client.
* Deposition of amount by the client.
* Faster balance enquiry.
* Instant money transfer.
* Payment of bills

**CHAPTER 2**

**LITERATURE REVIEW**

**1. Catalin Lupu: Vasile-Gheorghita Gaitan; Valeiru Lupu: Security enhancement of internet banking applications by using multimodal biometrics, Applied Machine Intelligence and Informatics (SAMI), 2015 IEEE 13th International Symposium**

This paper focuses on the enhancement of security features in online banking using multimodal biometrics for login and other operations. The login and signature security vary from user/static password authentication to certificates and tokens. Also, biometrics is increasingly used in many parts of our lives, from biometric passport, airport authentication and control access. It is easier and safer to login to internet banking. Also, signing an order will be more secure by using a fingerprint than a code generated by a token. A combination of these two authentication methods will lead to a visible security enhancement, too. The fingerprint can be used for two purposes: to open a token device and/or login to the internet banking application or sign an order. This paper will present a secure internet banking operations model by using biometrics.

# 2. FN Mahmadi; ZF Zaaba; A Osman: Computer Security Issues in Online Banking- An Assessment from the Context of Usable Security,

[**IOP Conference Series: Materials Science and Engineering**](http://iopscience.iop.org/journal/1757-899X)**,**[**Volume 160**](http://iopscience.iop.org/volume/1757-899X/160)**,**[**Number 1**](http://iopscience.iop.org/issue/1757-899X/160/1)

This paper highlights the subject of online banking security in Malaysia, especially from the perspective of the end-users. The study is done by assessing human computer interaction, usability and security. An online survey utilising 137 participants was previously conducted to gain preliminary insights on security issues of online banking in Malaysia. Following from those results, 37 participants were interviewed to gauge deeper understanding about end-users perception on online banking within the context of usable security. The results suggested that most of the end-users are continuingly experiencing significant difficulties especially in relation to the technical terminologies, security features and other technical issues. Although the security features are provided to provide a shield or protection, users are still incapable to cope with the technical aspects of such implementation.

**3. Janardan Choubey; Bhaskar Choubey: Secure User Authentication in Internet Banking: A Qualitative Survey, International Journal of Innovation, Management and Technology, Vol. 4, No. 2, April 2013**

This paper presents a qualitative survey of user identification mechanisms being applied in online banking environments across the English speaking world is presented. By studying the Internet banking sites of most major banks in 7 countries, the paper reports the variations and calls for standardization of user credentials in these environments.

**4. Tong Xin; Ban Xiaofang: Online Banking Security Analysis based on STRIDE Threat Model, Inetrnational journal of Security and its Applications, Vol.8, No.2(2014), pp.271-282**

This paper refers important issues regarding how to evaluate the security threats of the online banking effectively, a system threat analysis is proposed, which improves the efficiency of the threat analysis greatly and also has good practicability. By applying this method to the online banking system threat analysis, we construct STRIDE threat model on the analysis of the key business data, and then we construct threat tree on the security threat by layer-by-layer decomposition. Thus it gives a detailed threat analysis of the online banking system. This security threat analysis has important significance for the online banking system security analysis and for revealing the threats that the online banking facing.

**5. Rajpreet Kaur Jassal; Ravinder Kumar Sehgal: Online Banking Security Flaws: A Study, International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 8, August 2013**

This paper discusses about the criminal attacks on online banking systems. As the number of customers using online banking increases, online banking systems are becoming more desirable targets for criminals to attack. To maintain their customers’ trust and confidence in the security of their online bank accounts, financial institutions must identify how attackers compromise accounts and develop methods to protect them. The unique aspect about security in banking industry is that the security posture of a bank does not depend solely on the safeguards and practices implemented by the bank, it is equally dependent on the awareness of the users using the banking channel and the quality of end-user terminals. This makes the task for protecting information confidentiality and integrity a greater challenge for the banking industry. This paper aims to explains about the reason behind the security breaches and the participation of both customers and the banks to enable the hackers or crackers to access others network. The present study aims to find various types of flaws in the security of online banking that results in loss of money of account holders and financial institutions. Security breaches are not only because of banks faults and banks inadequate polices but customers are equally responsible for it, because customers awareness regarding security is equally important .

**CHAPTER 3**

**PROBLEM DESCRIPTION**

In today's technological world, people feel comfortable with their activities computerized thus reducing manual work and completing tasks instantaneously. One major sector that people would like to get actions performed instantly is banking. This lead to the invention of online and mobile banking facilities and a number of technologies paved the way for the online banking system development. But still security is a major issue in online banking and day-by-day there is a need of enhancing the security features in the online banking applications. One example of security threat is SQL injection which is a type of hacking where an unauthorized person can access a website and get details or make changes illegally.

Another issue with most web applications is that they are tightly coupled in which any changes to be made in the operations must be reflected in the source code to get it reflected in the operations. To overcome this websites or web applications serving the purpose must be converted into loosely coupled applications where a change to be made can be implemented in a single bean and there is no need of making changes in the source code.

**CHAPTER 4:**

**METHODOLOGY:**

**4.1 Eclipse**

Eclipse is an open source platform that helps a software developer to create a customized development environment (IDE) from plug-in components built by Eclipse members. Eclipse provides a program (UI) model for operating with tools.  It is designed to run on multiple operating systems while providing robust integration with each underlying OS.  Plug-ins can program to the Eclipse portable APIs and run unchanged on any of the supported operating systems

**About JAVA:**  **Features JDK 1.7**

**Platform Independent:**

The concept of Write-once-run-anywhere (known as the Platform independent) is one of the important key feature of java language that makes java as the most powerful language. Not even a single language is idle to this feature but java is closer to this feature. The programs written on one platform can run on any platform provided the platform must have the JVM.

**Simple:**

There are various features that make the java as a simple language. Programs are easy to write and debug because java does not use the pointers explicitly. It is much harder to write the java programs that can crash the system but we can not say about the other programming languages. Java provides the bug free system due to the strong memory management. It also has the automatic memory allocation and de-allocation system.

**Object Oriented:**

To be an Object Oriented language, any language must follow at least the four characteristics.

* Inheritance  : It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.
* Encapsulation: It is the mechanism of combining the information and providing the abstraction.
* Polymorphism:  As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.
* Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

As the languages like Objective C, C++ fulfills the above four characteristics yet they  are not fully object oriented languages because they are structured as well as object oriented languages. But in case of java,  it is a fully Object Oriented language because object is at the outer most level of data structure in java. No stand alone methods, constants, and variables are there in java. Everything in java is object even the primitive data types can also be converted into object by using the wrapper class.

**Robust:**

Java has the strong memory allocation and automatic garbage collection mechanism. It provides the powerful exception handling and type checking mechanism as compare to other programming languages. Compiler checks the program whether there any error and interpreter checks any run time error and makes the system secure from crash. All of the above features makes the java language robust.

**Distributed:**

The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these protocols and can get access the files from any remote machine on the internet rather than writing codes on their local system.

**Portable:**

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also have the standard data size irrespective of operating system or the processor. These features make the java as a portable language.

**Dynamic:**

While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.

**Secure:**

Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk. Java uses the public key encryption system to allow the java applications to transmit over the internet in the secure encrypted form. The byte code Verifier checks the classes after loading.

**Performance:**

Java uses native code usage, and lightweight process called threads. In the beginning interpretation of byte code resulted the performance slow but the advance version of JVM uses the adaptive and just in time compilation technique that improves the performance.

**Multithreaded:**

Java is also a multithreaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer.    
 Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other thread. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs. This is how Multithreading works in Java which you will soon come to know in details in later chapters.

**Interpreted:**

We all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code.   
The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program.   
The versatility of being **platform independent** makes Java to outshine from other languages. The source code to be written and distributed is platform independent.    
Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.

**Architecture Neutral:**

The term architectural neutral seems to be weird, but yes Java is an architectural neutral language as well. The growing popularity of networks makes developers think distributed. In the world of network it is essential that the applications must be able to migrate easily to different computer systems. Not only to computer systems but to a wide variety of hardware architecture and operating system architectures as well.  The Java compiler does this by generating byte code instructions, to be easily interpreted on any machine and to be easily translated into native machine code on the fly.The compiler generates an architecture-neutral object file format to enable a Java application to execute anywhere on the network and then the compiled code is executed on many processors, given the presence of the Java runtime system.Hence Java was designed to support applications on network. This feature of Java has thrived the programming language.

**ABOUT : JDK:**

The **Java Development Kit** (**JDK**) is a [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) product aimed at [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) developers. Since the introduction of Java, it has been by far the most widely used Java [SDK](http://en.wikipedia.org/wiki/Software_development_kit). On [17 November](http://en.wikipedia.org/wiki/November_17) [2006](http://en.wikipedia.org/wiki/2006), Sun announced that it would be released under the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL), thus making it [free software](http://en.wikipedia.org/wiki/Free_software). This happened in large part on [8 May](http://en.wikipedia.org/wiki/May_8) [2007](http://en.wikipedia.org/wiki/2007)[[1]](http://en.wikipedia.org/wiki/Java_Development_Kit#cite_note-0#cite_note-0) and the source code was contributed to the [OpenJDK](http://en.wikipedia.org/wiki/OpenJDK).

The primary components of the JDK are a selection of programming tools, including:

* java – The [loader](http://en.wikipedia.org/wiki/Loader) for Java applications. This tool is an interpreter and can interpret the class files generated by the [javac](http://en.wikipedia.org/wiki/Javac) compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, is no longer provided with Sun JDK.
* [javac](http://en.wikipedia.org/wiki/Javac) – The [compiler](http://en.wikipedia.org/wiki/Compiler), which converts source code into [Java bytecode](http://en.wikipedia.org/wiki/Java_bytecode)
* jar – The archiver, which packages related class [libraries](http://en.wikipedia.org/wiki/Library_(computer_science)) into a single [JAR file](http://en.wikipedia.org/wiki/Jar_(file_format)). This tool also helps manage JAR files.
* [javadoc](http://en.wikipedia.org/wiki/Javadoc) – The documentation generator, which automatically generates documentation from [source code](http://en.wikipedia.org/wiki/Source_code) comments
* jdb – The [debugger](http://en.wikipedia.org/wiki/Debugger)
* javap – The class file disassembler
* appletviewer – This tool can be used to run and debug Java applets without a web browser.
* javah – The C header and stub generator, used to write native methods
* extcheck – This utility can detect JAR-file conflicts.
* apt – The annotation processing tool
* jhat – (Experimental) Java heap analysis tool
* jstack – (Experimental) This utility prints Java stack traces of Java threads.
* jstat – (Experimental) [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) statistics monitoring tool
* jstatd – (Experimental) jstat daemon
* jinfo – (Experimental) This utility gets configuration information from a running Java process or crash dump.
* jmap – (Experimental) This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump.
* idlj – The IDL-to-Java compiler. This utility generates Java bindings from a given IDL file.
* policy tool – The policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources
* [VisualVM](http://en.wikipedia.org/wiki/VisualVM) – visual tool integrating several command line JDK tools and lightweight performance and memory profiling capabilities

The JDK also comes with a complete [Java Runtime Environment](http://en.wikipedia.org/wiki/Java_Runtime_Environment), usually called a *private* runtime. It consists of a [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) and all of the class libraries that will be present in the production environment, as well as additional libraries only useful to developers, such as the [internationalization](http://en.wikipedia.org/wiki/Internationalization_and_localization) libraries and the [IDL](http://en.wikipedia.org/wiki/Interface_description_language) libraries.

Also included are a wide selection of example programs demonstrating the use of almost all portions of the [Java API](http://en.wikipedia.org/wiki/Java_API).

**4.2 Oracle SQL**

Oracle SQL Developer is an incorporated improvement instrument that disentangles the advancement and administration of Oracle Database in cloud stage. SQL Developer offers an entire end-to-end improvement of uses, a workspace for running questions and scripts, a Database Admin reassure for dealing with the database, a report interface, a total information displaying arrangement, and a relocation stage for moving your outsider databases to Oracle. SQL database has been utilized to store the subtle elements of the worker and it does the way toward embeddings, recovering and refreshing the records of a representative.SQL gives a simple, exquisite, preferment design for getting to, characterizing, and looking after information

**4.3 Spring**

The Spring Framework is a lightweight framework which is used for building enterprise-ready applications. As the spring framework is flexible it allows to use only the parts which are required and those parts are not needed to bring to the rest. The framework also supports remote access to logic of the code through some web services and also supports transaction management. It also provides many ways to store the persistent data. The framework consists of an container called IOC container, using this one can integrate any framework on the top and also can use the JDBC abstraction layer or Hibernate integration code

Full-featured framework and enables to integrate the Aspect Oriented Programing ([AOP](https://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#aop-introduction)) transparently into your software. As the spring framework is dependent on java platform, it provides a platform for developing java applications.

**4.4 Restful API**

Restful API which is called as Representational state transfer(REST) or Restful web services are one of the way which coalesce the computer system on the internet. REST web services provide access and also help to change the textual representations of web sources using some stateless protocols and operations. Many other forms of web services also exist like Web Service Definition language (WSDL) and Simple Object Access Protocol(SOAP), they provide their own set of arbitrary operations. In a RESTful Web service, requests made to a resource  [URL](https://en.wikipedia.org/wiki/URI" \o "URI) will receive a response which might be in that may be in [XML](https://en.wikipedia.org/wiki/XML), [HTML](https://en.wikipedia.org/wiki/HTML), [JSON](https://en.wikipedia.org/wiki/JSON) etc.. The response body provides if any alteration is made to the already stored record and also provides methods like GET, PUT, POST,DELETE and also provides some hyperlinks to the other resources which are related or collection of resources. By using  [HTTP](https://en.wikipedia.org/wiki/HTTP), which is one of the most common type of operations available which are predefined by the [HTTP verbs](https://en.wikipedia.org/wiki/HTTP_verbs) are GET, POST, PUT, DELETE etc. By using stateless protocols REST systems are liable for fast performance, reliability and flexibility and as they reuse the components it is easy to manage and update even when the application is running.

**CHAPTER 5**

**SOFTWARE REQUIREMENT SPECIFICATION**

**5.1 Software requirements**

The following are the minimum requirements that the target web application is expected to satisfy for smooth running of this application.

Operating System: Windows 2007 or above

Eclipse: 3.4 or greater

HTML 5

CSS 3

**5.2 Hardware Requirements**

The following are the minimum hardware requirements that the target system is expected to satisfy for running this program.

Processor: 1 GHz or Higher

RAM: 1GB or higher

Storage Space: 50MB free space

**5.3 Development Specifications**

Technologies used: Java (Eclipse), Spring

IDE used: Eclipse, Maven Platform

**5.4 Functional requirements**

The following functional requirements were specified

* The application checks if the entered username and password are valid.
* The application will allow the users to view the current balance, previous transaction history, make payments/transfers and pay bills
* This application allows the manager or an authorized employee of the bank to view the details of any accountholder and make transactions and changes in the account

**5.5 NON FUNCTIONAL REQUREMENTS**

**5.5.1 Performance requirements**

The user should be able know the next step in the process with certain prompts. This makes the application user friendly. Care should be taken as to observe that no column is filled with erroneous or ghost data. While signing up to the application, all the details get stored in the database. The application should be user friendly and it should be faster to access.

**5.5.2 Operational Requirements**

The application should be ready to be deployed in other applications with limited and easy customizations. Such as able to run on any browser and data and can be integrated as a component or plug-in. This makes the application that is to be developed behave as a component in the real world.

**5.5.3 Alert Messages**

The user should know the process and must be alerted with accurate error messages and the error codes must be decoded or translated in such a way that the user should know what exactly the error is and what is supposed to be done by the user for the process to be completed successfully.

**CHAPTER 6**

**PRESENT WORK**

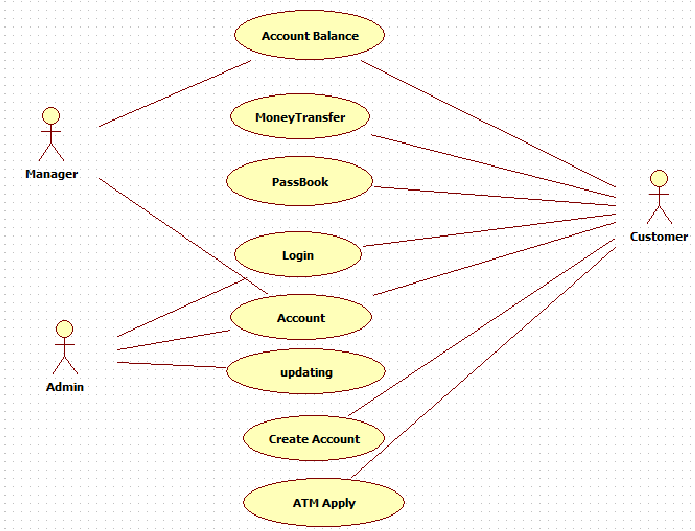
****

Figure 6.1 Use Case Diagram for Online Banking

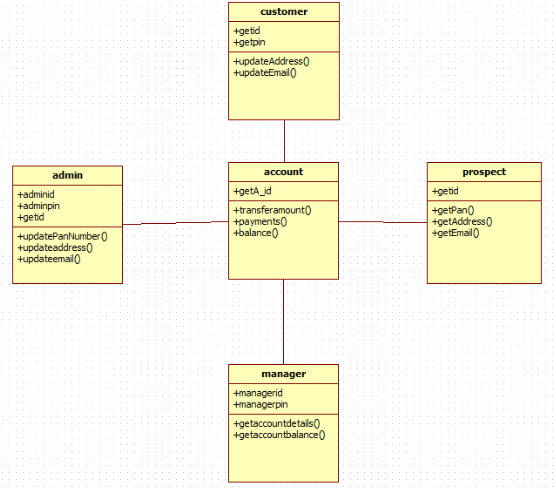
****

Figure 6.2 Class Diagram for Online Banking

**CHAPTER 7**

**RESULTS AND DISCUSSION**

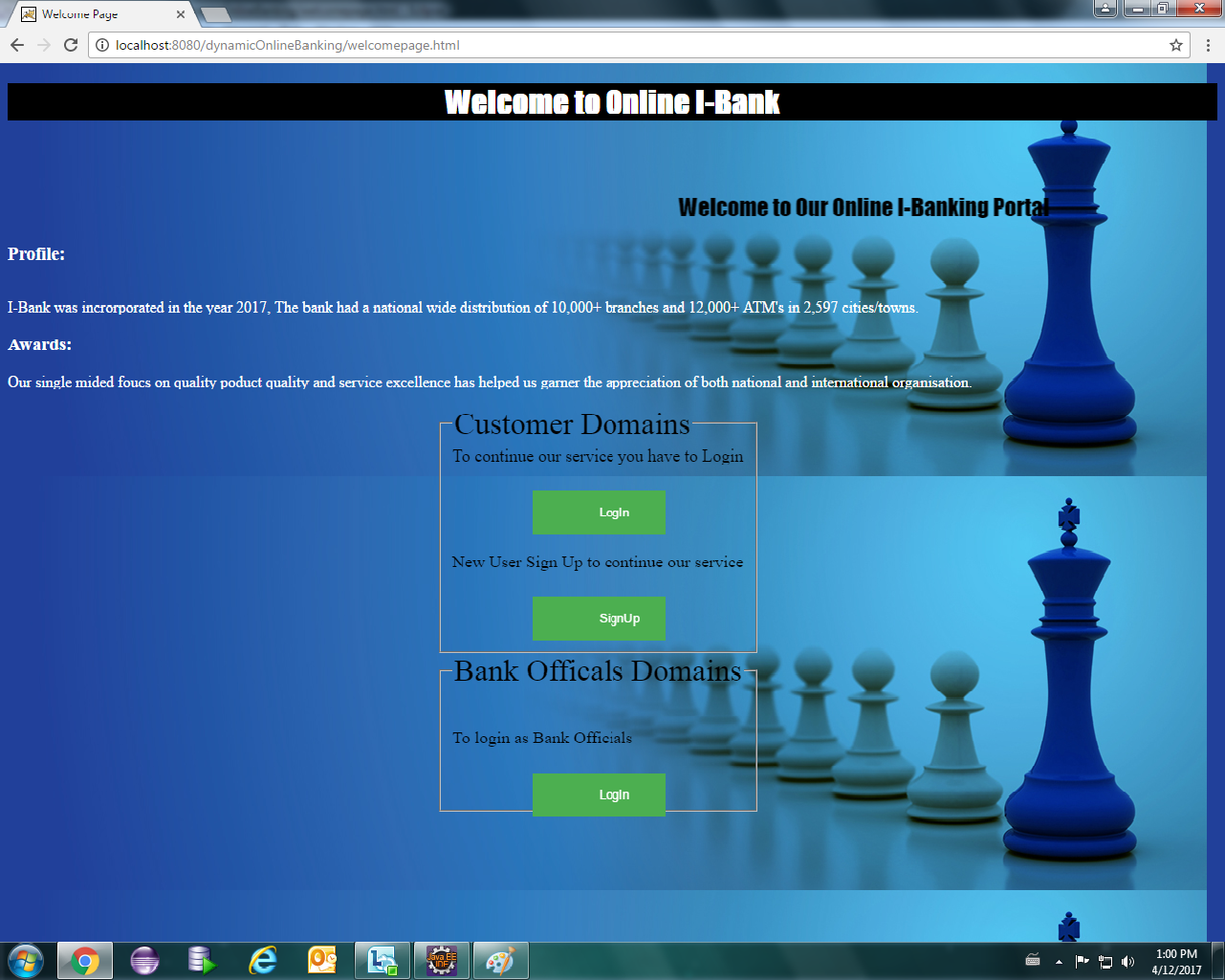
****

Figure 7.1 Online bank welcome page

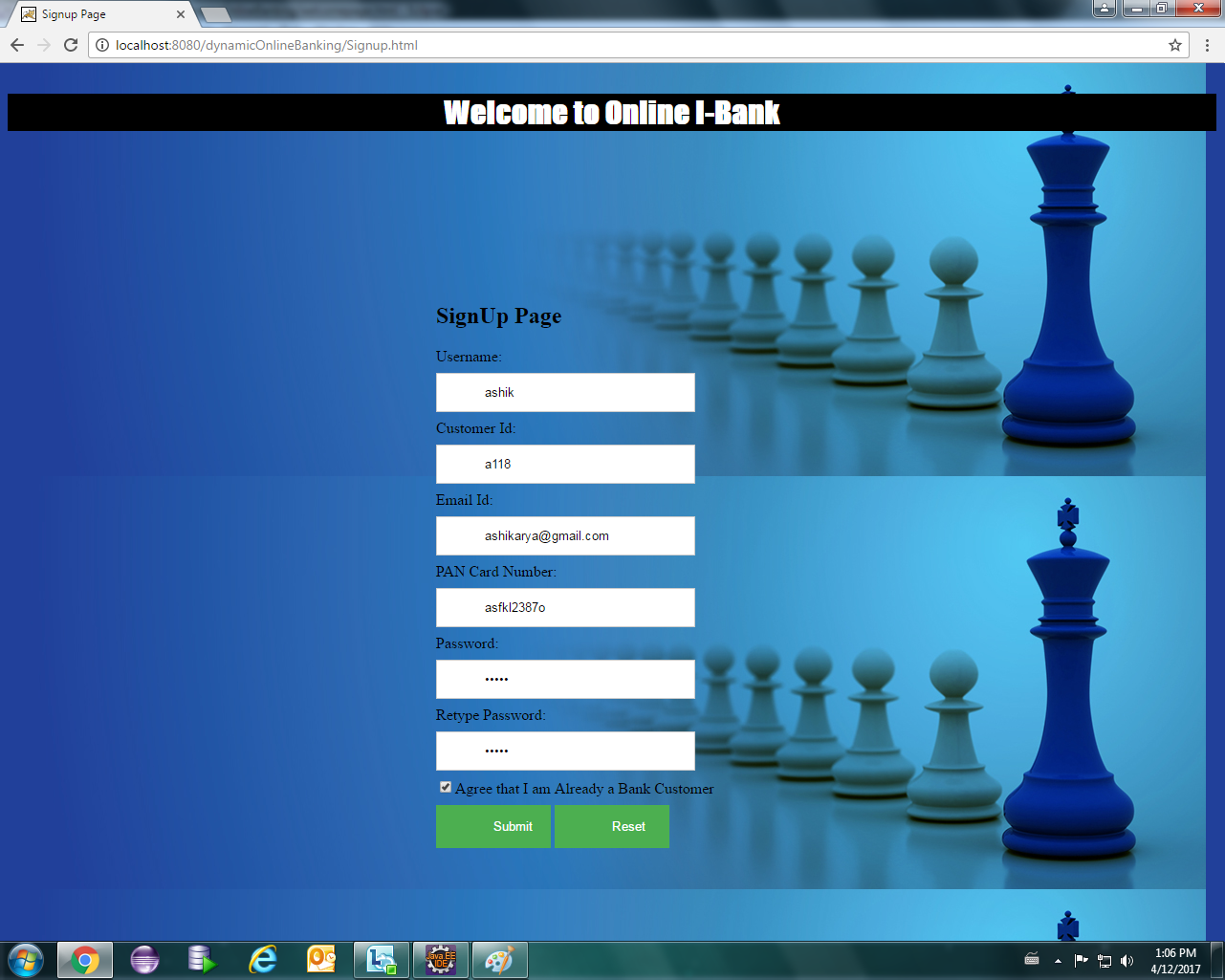
****

Figure 7.2 Sign Up page

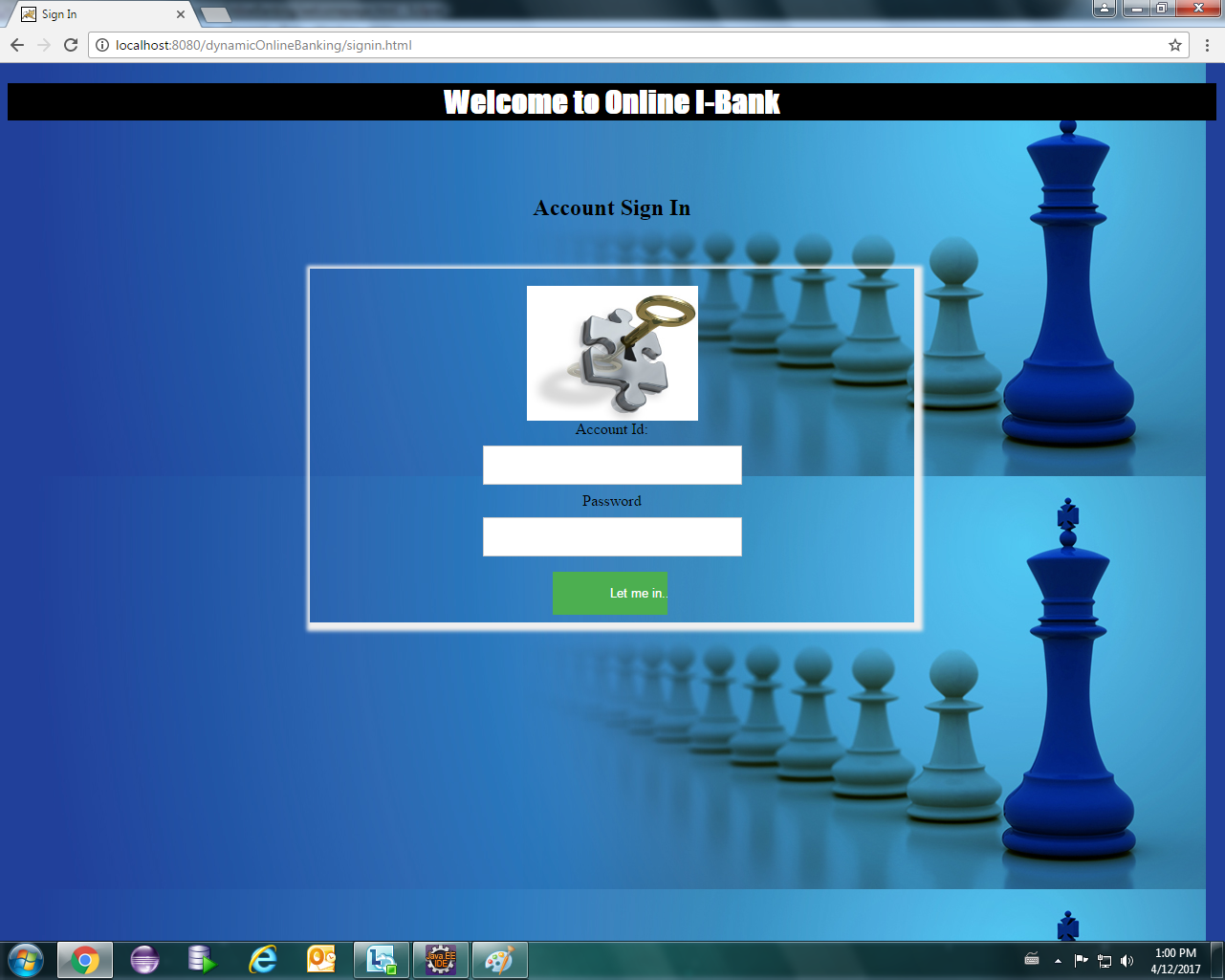
****

Figure 7.3 Sign In page

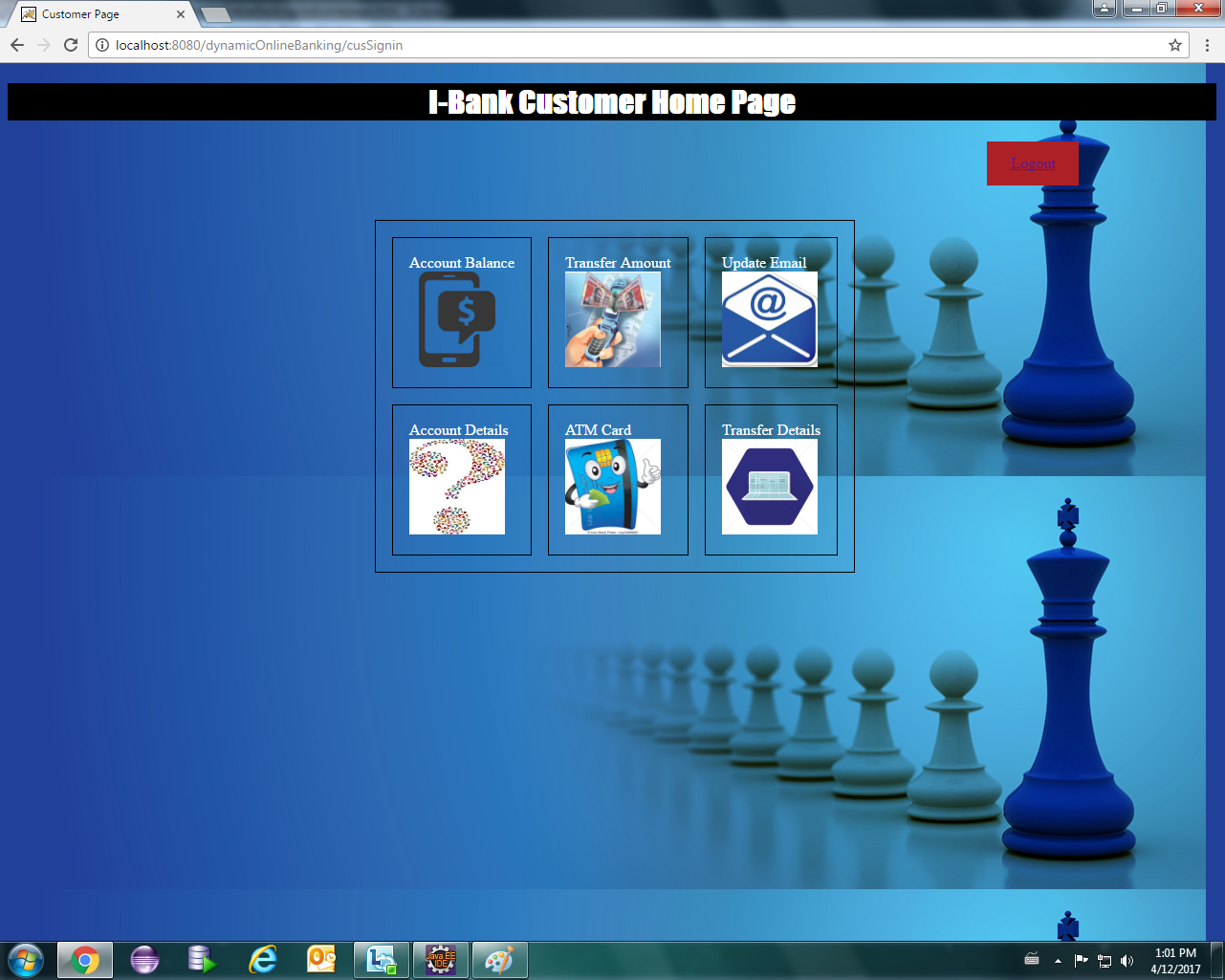
****

Figure 7.4 Online bank home page

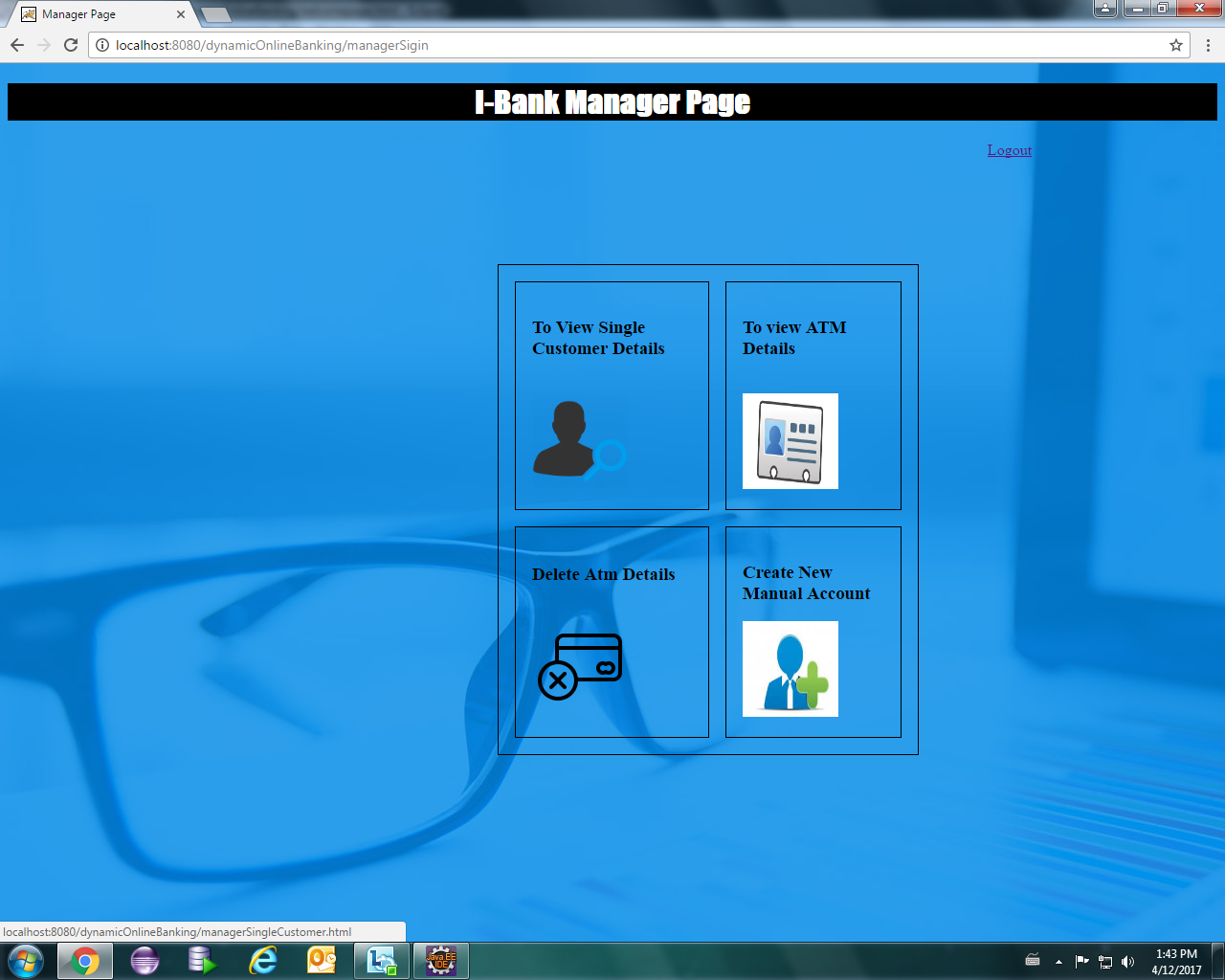
****

Figure 7.5 Manager home page

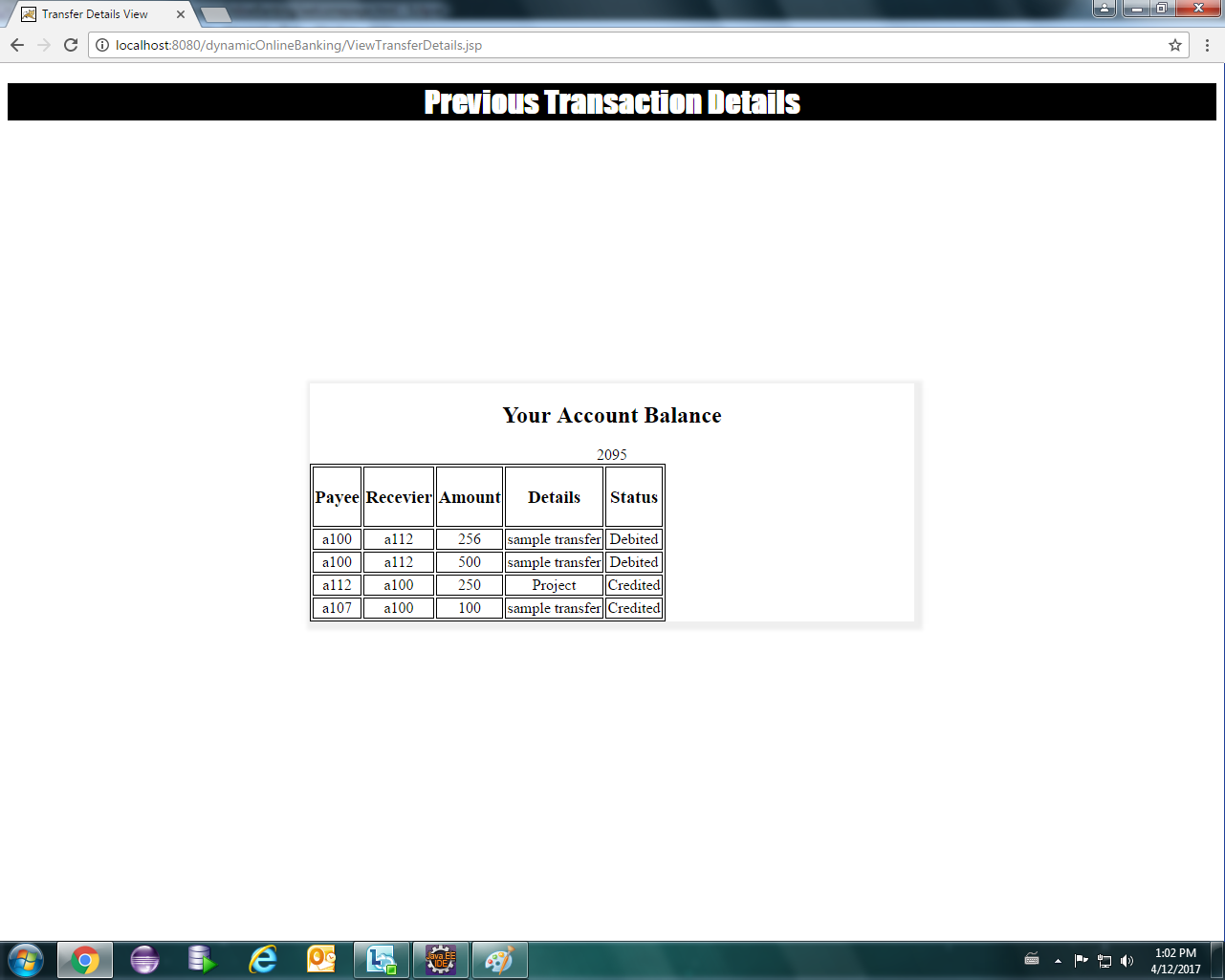
****

Figure 7.6 Previous transactions details

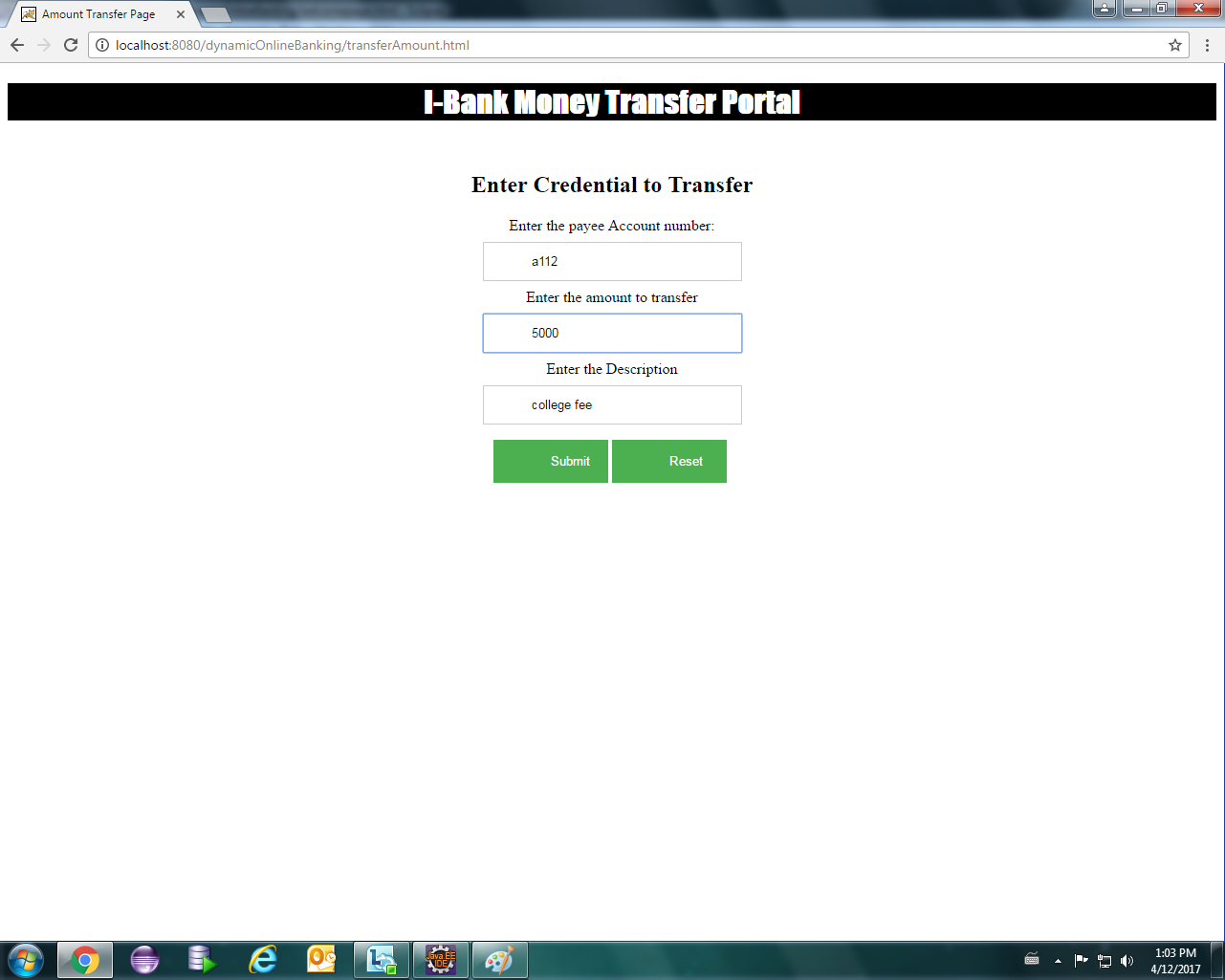
****

Figure 7.7 Money Transfer Portal

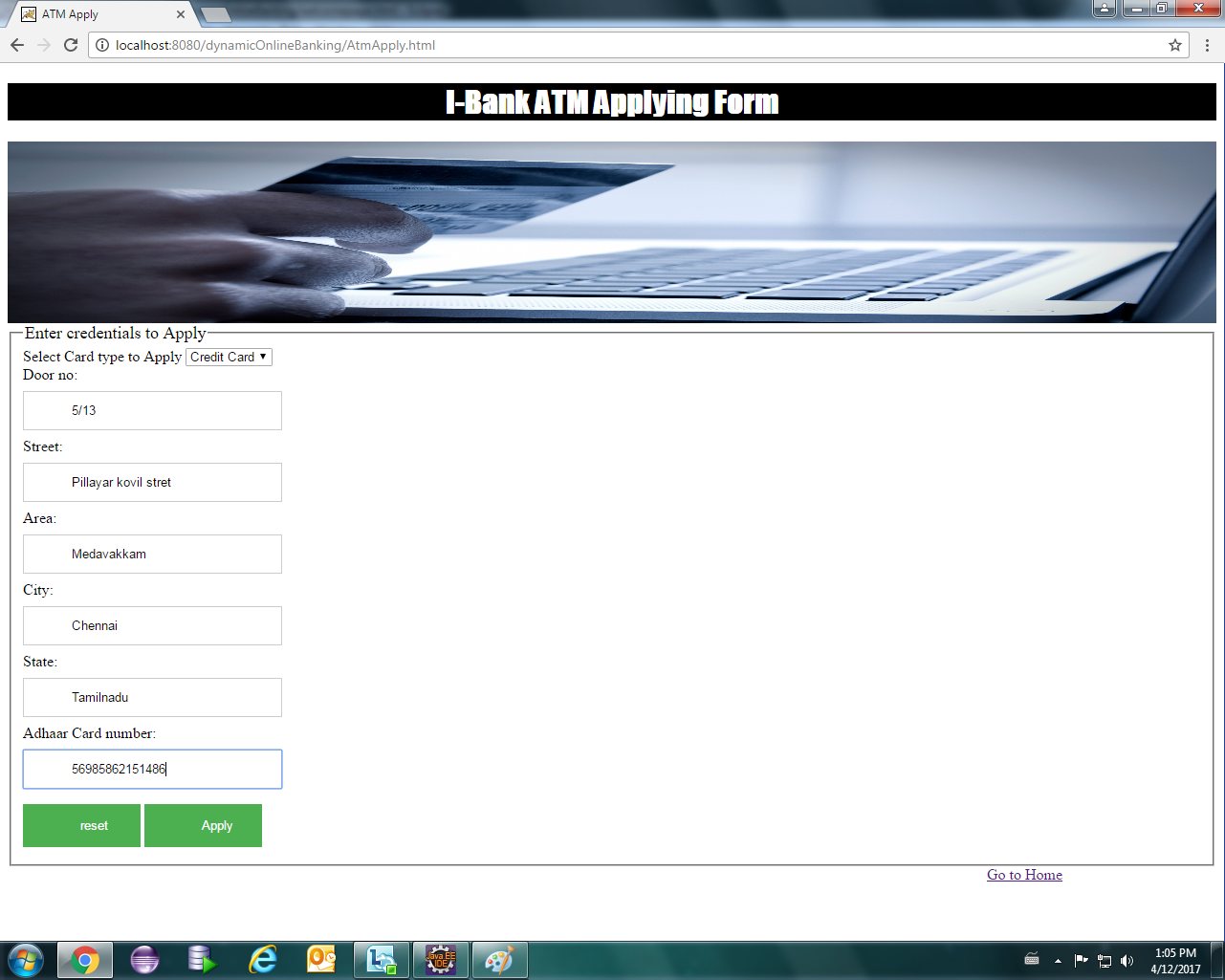
****

Figure 7.8 ATM applying form

****

Figure 7.9 Update Email

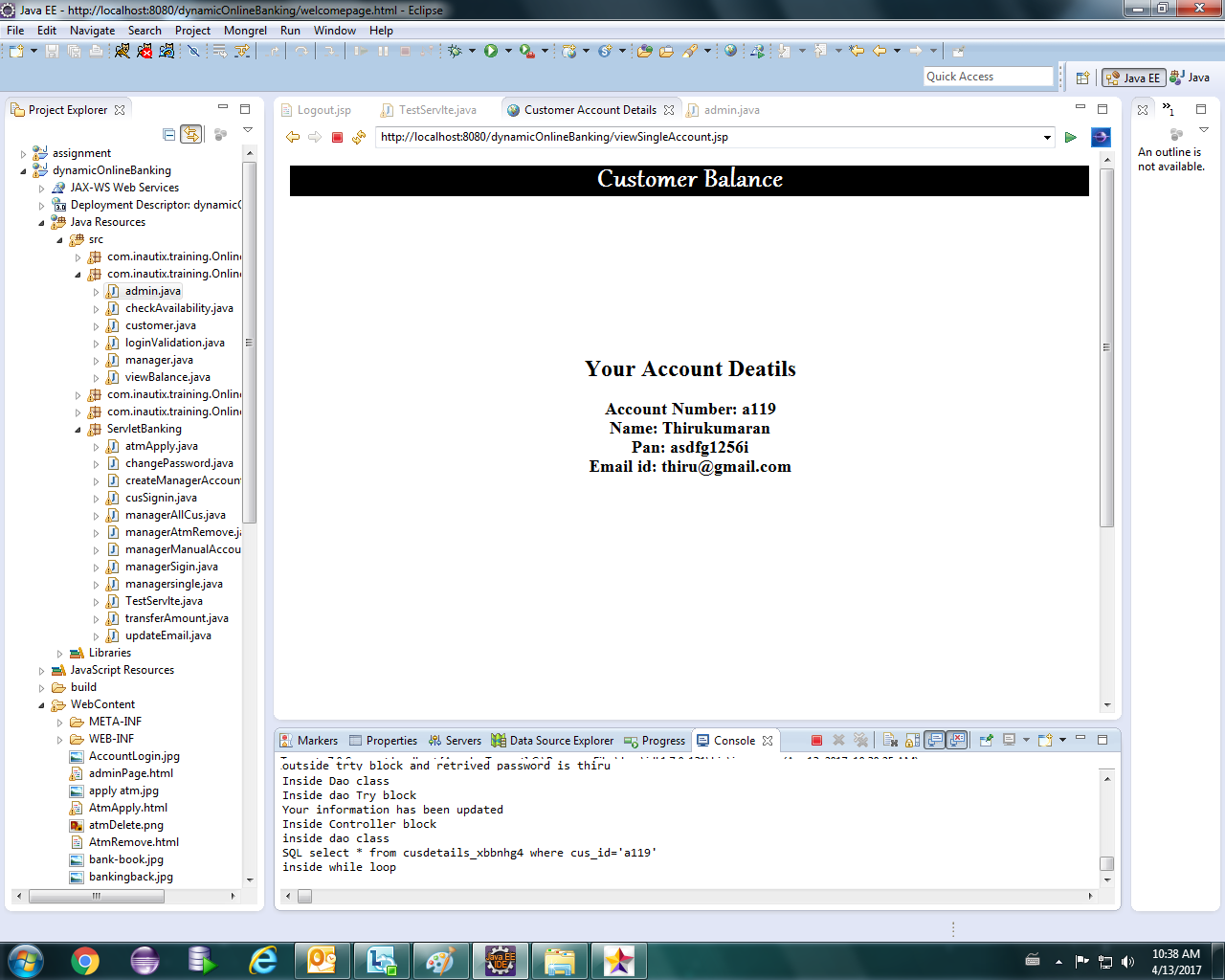
****

Figure 7.10 Customer account balance details

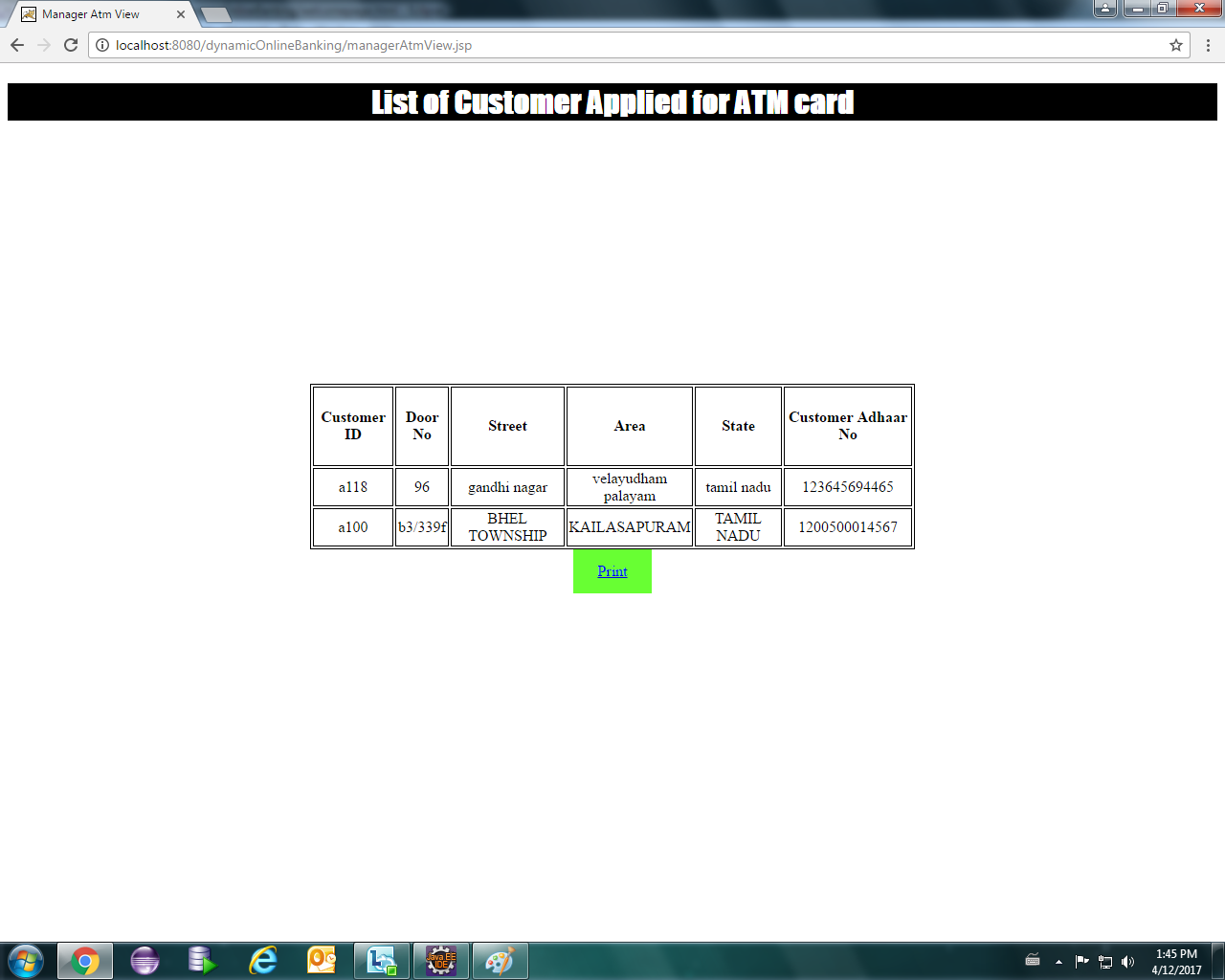
****

Figure 7.11 List of customer applied for ATM card

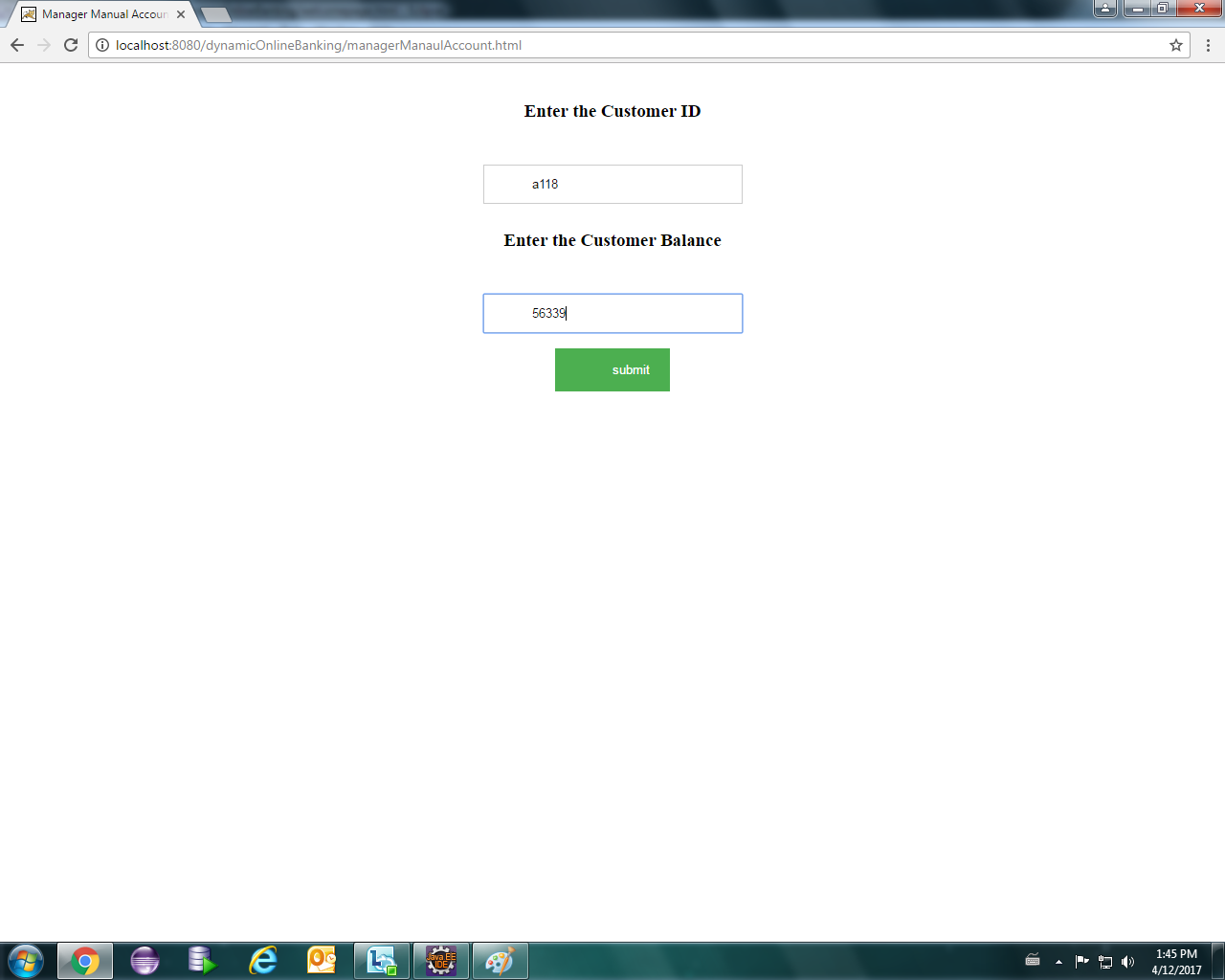
****

Figure 7.12 Account balance manager view

** **

Figure 7.13 Backend servlet output

**CHAPTER 8**

**CONCLUSION AND FUTURE WORK**

The application developed has been successfully developed and subjected to various tests and measures that include factors like balance enquiry, funds transfer, updating details, transaction history and a few more operations. The application is loosely coupled which allows changes in the operations be made easily. The User Interface is user friendly and simple. Dynamical updation of the details has been performed. This application can be furthermore enhanced with payment gateway support for interbank transactions. Integration is being done by adding new tabs in the application using spring framework. In future, more security features can be incorporated in the application to make it more secure and trustworthy.

**REFERENCES**

1. Catalin Lupu: Vasile-Gheorghita Gaitan; Valeiru Lupu: Security enhancement of internet banking applications by using multimodal biometrics, Applied Machine Intelligence and Informatics (SAMI), 2015 IEEE 13th International Symposium

# FN Mahmadi; ZF Zaaba; A Osman: Computer Security Issues in Online Banking- An Assessment from the Context of Usable Security,

1. [IOP Conference Series: Materials Science and Engineering](http://iopscience.iop.org/journal/1757-899X), [Volume 160](http://iopscience.iop.org/volume/1757-899X/160), [Number 1](http://iopscience.iop.org/issue/1757-899X/160/1)
2. Janardan Choubey; Bhaskar Choubey: Secure User Authentication in Internet Banking: A Qualitative Survey, International Journal of Innovation, Management and Technology, Vol. 4, No. 2, April 2013
3. Tong Xin; Ban Xiaofang: Online Banking Security Analysis based on STRIDE Threat Model, Inetrnational journal of Security and its Applications, Vol.8, No.2(2014), pp.271-282
4. Rajpreet Kaur Jassal; Ravinder Kumar Sehgal: Online Banking Security Flaws: A Study, International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 8, August 2013

**WEB REFERENCES**

1. [www.inautix.e-box.co.in](http://www.inautix.e-box.co.in/)
2. [www.codeacademy.com/learn/web](http://www.codeacademy.com/learn/web)
3. [www.codeacademy.com/learn\_the\_command\_line](http://www.codeacademy.com/learn_the_command_line)
4. [www.nptel.ac.in/courses/106106127/1](http://www.nptel.ac.in/courses/106106127/1)
5. [www.udemy.com/spring\_tutorial\_for\_beginners](http://www.udemy.com/spring_tutorial_for_beginners)
6. [www.beginnersbook.com/java\_tutorial](http://www.beginnersbook.com/java_tutorial_for_beginners)
7. [www.javatpoint.com](http://www.javatpoint.com/)
8. [www.tutorialspoint.com](http://www.tutorialspoint.com/)