**RADIO FREQUENCY IDENTIFICATION BASED TOLL COLLECTION SYSTEM**

## 

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that this is a bonafide record of the project work doneon  
 **“Radio Frequency Identification Based Toll Collection system”**

BY

Zeiba Baig (Seat No. 0683)

Priyanka Bandekar (Seat No. 0684)

Joshua Fernandes (Seat No. 0693)

Shreya Joshi (Seat No. 0700)

Siya Naik (Seat No. 0713)

VeverlyVaz (Seat No. 0752)

Yashin Nagvenkar (Seat No. 0754)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Prof. Fiona Coutinho) (Prof.Amey J. ShenviKhandeparkar)

(Internal Guide) (Internal Co-Guide)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Prof. KavitaAsnani) (Dr. Luis C. Mesquita)

(HOD-IT) (Principal)

Date:

Place:

## 

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**EXAMINER SIGNATURE SHEET**

This is to certify that the following students successfully completed their project work for the year 2013-2014 done on  
 **“Radio Frequency Identification Based Toll Collection system”**

BY

Zeiba Baig (Seat No. 0683)

Priyanka Bandekar (Seat No. 0684)

Joshua Fernandes (Seat No. 0693)

Shreya Joshi (Seat No. 0700)

Siya Naik (SeatNo. 0713)

Veverly Vaz (Seat No. 0752)

Yashin Nagvenkar (Seat No. 0754)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Internal Examiner) (External Examiner)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Prof. KavitaAsnani) (Dr. Luis C. Mesquita)

(HOD-IT) (Principal)

Date:

Place:

**TABLE OF CONTENTS**

ACKNOWLEDGEMENT 05

INTRODUCTION 06

1.1PRESENT SCENARIO 07

1.2 PROBLEM DEFINATION 07

1.3 SCOPE 08

TECHNOLOGIES USED 09

2.1 RFID 10

2.2 SOFTWARE 11

* Front End
* Back end

2.3HARDWARE 16

DESIGN 17

3.1 USE CASE DIAGRAM 18

3.2 DATA FLOW DIAGRAM 19

3.3 ENTITY RELATIONSHIP DIAGRAM 21

IMPLEMENTATION 23

4.1 TABLE DESIGN 24

4.2 FORM DESIGN 27

* Admin Login
* User/Customer Login
* Toll Deduction Application

TESTING 37

5.1 TEST CASES 39

FUTURE SCOPE 41

BIBLIOGRAPHY 42

**ACKNOWLEDGEMENT**

We take this opportunity to whole heartedly thank all those who were instrumental in helping us achieve our goals.

We sincerely express our gratitude to our internal project guides Prof. Fiona Coutinho, Prof. Amey J. ShenviKhandeparkarand Prof. Nadia Barettofor their encouragement, guidance and valuable help without which this project would not have been successful.

We owe grateful thanks to Principal Dr. Luis C. Mesquita and Prof. KavitaAsnanifor their co-operation and support.

We would also like to express our gratitude to our project co-ordinatorProf. Nadine Dias, Prof. MeghaAinapurkar and also the lab Assistants Mr. Michael , Mrs. Linette, Mrs. Faye and Mr. Ryan.

We would like to thank our parents for their constant support and guidance.

Above all we would like to thank the almighty for giving us the courage, strength and determination to finish this project successfully.

Thank you one and all…

***CHAPTER 1***

**INTRODUCTION**

**INTRODUCTION**

**1.1 PRESENT SCENARIO**

Interacting with various citizens specially with frequent road travelers and through our very own experience we took into account the inconvenience caused due to manual toll collection and extended our interaction to get the opinions of the RTO department too hence observed the working of the manual toll as :

* The toll booth functions in an entirely manual manner.
* Once a vehicle arrives at the toll, it has to halt for a subsequent amount of time.
* The toll master then shall collect the prescribed fee with respect to the vehicle.
* On payment of the fee, after required calculation on the amount received if successful shall open the gate.
* A lot of congestion is caused with long waiting lines.
* With a need to achieve fast service it frequently leads to human computation errors.
* User/vehicle owner has no opportunity to view his details /records.
* Overall it contributes to the poor functioning of the toll collection system with inconvenience caused.

**1 .2 PROBLEM DEFINATION**

RFID Based Toll Collection System aims to build up well regularized and efficient software for toll collection for the state of Goa, which shall aim at reducing processing time and minimize manual errors thus increasing the highway capacity (efficient functioning).

By implementing the RFID TOLL COLLECTION SYSTEM in real world we aim at achieving well regularized and organized toll collection with short waiting time thus leading to time saving and reducing the frustration of travellers.

In addition the processing time is reduced efficiently along with absence of human errors.

On achieving the above two objectives it shall also lead to increasing highway capacity and fuel economy.

**1.3 SCOPE**

The project consists of two users:

* The Client :
* sign up and register account
* view account details
* update and modify specific fields
* recharge voucher
* tapping the tag
* The Admin:
* register new users
* view user details
* view transactions
* add new vouchers
* send text message
* auto gate opening

***CHAPTER 2***

**TECHNOLOGIES USED**

**TECHNOLOGIES USED**

**2.1 RADIO FREQUENCY IDENTIFICATION:**

* Radio Frequency Identification (RFID) is an automated data capture technology that can be used to electronically identify, track, and store information contained on a tag that is attached to or embedded in an object, such as a product, case or pallet.
* Silent Commerce focuses on using advanced tagging, sensor and actuator technologies to make everyday objects intelligent and interactive. It’s called “silent” because objects communicate and conduct commerce without human interaction.
* RFID-silicon-based electronic identification tags, consisting of a tiny processor, memory, antenna.
* RFID tags can be read and written wirelessly and can be made cheap, without a battery

**Merits of RFID:**

* Reduce travel time.
* Reduce wait time at toll booths.
* Toll collection is automated, hence human error is eliminated.
* Reduce in auto emission.
* Increase in fuel economy.
* Increase in highway capacity.

**2.2 SOFTWARE**

* **FRONT END:**
* HTML

HTML or Hyper-text Mark-up Language is a globally accepted programming language for formatting web pages. In today's world, it is commonly used along with JavaScript and Cascading Style Sheets (CSS) to give web pages the look and feel we desire. Through HTML, the look and appearance of images, links, headings, text, page layout and just about every element of a web page can be formatted. HTML is the predominant programming language for creating web pages. It is also the most optimal for most small and growing businesses that do not really need advanced functionality on their website.

**Merits of HTML:**

* HTML is easy to use and understand
* Almost anyone in the web development business would know HTML – be it a freelancer or a large agency. If at any point in time you need to hire the services of a different web design firm or professional for making changes or updates to your website, it would be relatively easy to find cost-effective and affordable solution providers who can make the changes you need to your website.
* All browsers support HTML
* Almost – if not all – browsers support HTML. Certainly more browsers support HTML than any other web programming language. As a result, when you build a website using HTML, it would show up on most browsers around the world, as long as the programmer takes care to optimize the website for the most commonly used browsers. Optimizing an HTML based website for browser compatibility is neither difficult nor complex.
* HTML and XML syntax is very similar
* Today, XML is increasingly being used for data storage. The similarity of syntax between HTML and XML means that it is easier and seamless working between the two platforms.
* HTML is free
* A major advantage of HTML is that it is free. You do not need any software for HTML, no plug-ins are needed and it means that you can save considerably on your website development cost. Even with open source content management systems, all the plug-ins that you may need are not always free.
* Most development tools support HTML
* Whether it is FrontPage, Dreamweaver or any other programming tool, there are more web development tools that allow you to create HTML based websites, than any other web programming language.
* HTML is most search engine friendly
* Of all the web programming languages, HTML is the most search engine friendly. Creating SEO compliant websites using HTML is significantly easier than any other programming language. HTML causes the least SEO complications and provides the greatest flexibility when trying to build an SEO compliant website. As long as you have taken care to ensure your HTML code is clean and validated, an HTML website is easiest to read and access for search engine crawlers. This reduces crawling time and improves page load time, helping your website perform better in search results.
* NetBeans IDE
* An **Integrated Development Environment** is computer software to help computer programmers develop software

**NEED FOR IDE:**

* IDE abstracts the configuration necessary to piece together various utilities in one unit, which could ease the learning of a language, and increases developer productivity.
* Most IDEs today have GUI modelling utilities that simplify the development of UIs, which is critical for commercial software today.
* It all started as a student project called Xelfi.The Goal was to write a Delphi- like Java IDE in Java for the first time.The original plan was to develop network-enabled JavaBeans components, hence the name. But coming out of the spec for enterprise changed the plans.Sun decided it needs a more powerful Java development tool, and the rest is history.
* NetBeans was now a fastand fully-featured Integrated Development Environment (IDE) with support for Java.
* Compliant applications for accelerating development across all major OS platforms.
* Provides an open source, high performance, modular, extensible, multi-platform Java IDE for GUI, mobile tools, Web, and Desktop applications.
* Written in java and therefore runs on every operating system that supports Java VM.

**Features:**

* **Environment:**Easily configured user interface and a modular architecture extensible with additional plugins.
* **Project System:**Support for multiple source roots, easy management of libraries, easily ported to other environments, all based on Apache Ant.
* **Enterprise Java Beans (EJB) Development:**Easy to create and deploy and import java beans.
* **Web Services Development:**Wizards for creating web services and web services clients, providing the basic (java/wsdl) code needed, and easy to use testing tools of existing web services.
* **Java 2 Platform, Micro Edition (J2ME) MIDP development:**Visual design editor with end-to-end support for enterprise applications.
* **Code Editor:** Syntax highlighting for Java, XML, HTML, CSS, JSP and IDL, full support of new JDK 1.5 features, live parsing/error marking, popup javadoc, code completion, and fast class importing.
* **Refactoring:**Renaming, changing and moving of various objects, field encapsulation and usage finding.
* **Award Winning Debugger:** Language independent debugger core, variable modification and watches, various breakpoints and “Fix and Continue” mechanism.
* **Web Development:** Web Application project type, Supports the J2EE 1.3 and 1.4 standards with web application build support based on Apache Ant.
* **GUI Builder:**Fully WYSIWYG designer with "Test Form" feature, extensible Component Palette pre-installed Swing and AWT components, showing a components tree and properties, automatic code generation and full JavaBeans support.
* **Version control Support:**Supports command lined VCS, supplying merging and diff tools and containing a built- in CVS client.
* **XML:** XML, DTD and CSS Text Editor and XML Productivity Tools Wizards to help user generate codes.
* GlassFish Server

**GlassFish** is an [open-source](http://en.wikipedia.org/wiki/Open-source_software)[application server](http://en.wikipedia.org/wiki/Application_server) project started by [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) for the [Java EE](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) platform and now sponsored by [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation). The supported version is called Oracle GlassFish Server. GlassFish is [free software](http://en.wikipedia.org/wiki/Free_software), [dual-licensed](http://en.wikipedia.org/wiki/Dual-licensing) under two [free software licences](http://en.wikipedia.org/wiki/Free_software_licence): the [Common Development and Distribution License](http://en.wikipedia.org/wiki/Common_Development_and_Distribution_License) (CDDL) and the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL) with the [classpath exception](http://en.wikipedia.org/wiki/GPL_linking_exception#The_classpath_exception).

* JAVA TECHNOLOGIES
  + JDBC Drivers
  + **JDBC** is a Java-based data access technology (Java Standard Edition platform) from Oracle Corporation. This technology is an API for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database.
  + JSP (Java Server Pages)
  + Java Development Kit (JDK)
* **BACK END:**
* PHP**:**

**PHP** is a [scripting](http://en.wikipedia.org/wiki/Scripting_programming_language) language designed to fill the gap between[SSI](http://en.wikipedia.org/wiki/Server_Side_Includes) (Server Side Includes) and [Perl](http://en.wikibooks.org/wiki/Perl), intended for the web environment. Its principal application is the implementation of web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C-like syntax, and its simplicity. PHP is currently divided into two major versions: PHP 4 and PHP 5, although PHP 4 is deprecated and is no longer developed or supplied with critical bug fixes. PHP 6 is currently under development.

Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "**P**HP: **H**ypertext **P**reprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications which can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL.

* MySql**:**

MySQL is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), and ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface)tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](http://en.wikipedia.org/wiki/Command_line) tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, [MySQL Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) is actively developed by Oracle, and is freely available for use.

The official [MySQL Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, [MySQL GUI Tools](http://en.wikipedia.org/wiki/MySQL_GUI_Tools). Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

MySQL Workbench is available in two editions, the regular [free and open source](http://en.wikipedia.org/wiki/Free_and_open_source_software) *Community Edition* which may be downloaded from the MySQL website, and the proprietary *Standard Edition* which extends and improves the feature set of the Community Edition.

MySQL is written in [C](http://en.wikipedia.org/wiki/C_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B). Its SQL parser is written in [YACC](http://en.wikipedia.org/wiki/Yacc), but it uses a home-brewed [lexical analyzer](http://en.wikipedia.org/wiki/Lexical_analysis). Many[programming languages](http://en.wikipedia.org/wiki/Programming_language) with language-specific [APIs](http://en.wikipedia.org/wiki/Application_programming_interface) include [libraries](http://en.wikipedia.org/wiki/Library_(computing)) for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's [Visual Studio](http://en.wikipedia.org/wiki/Visual_Studio) (languages such as [C#](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)) and [VB](http://en.wikipedia.org/wiki/Visual_Basic) are most commonly used) and the JDBC driver for Java. In addition, an [ODBC](http://en.wikipedia.org/wiki/ODBC) interface called [MyODBC](http://en.wikipedia.org/wiki/MyODBC) allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages) or [ColdFusion](http://en.wikipedia.org/wiki/Adobe_ColdFusion). The [HTSQL](http://en.wikipedia.org/wiki/HTSQL) – [URL](http://en.wikipedia.org/wiki/Uniform_resource_locator)-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

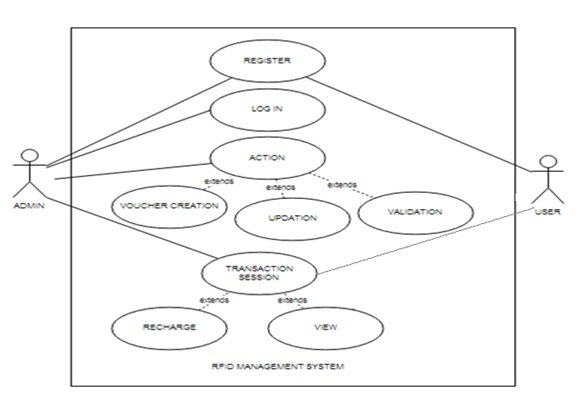
**2.3 HARDWARE**

* RFID READER **:**
* Readers, which are often called “interrogators”, are complementary to tags and can be as technically diverse as tags.
* A reader sends a pulse of energy “to the tag and listens for the tag’s response”.
* The tag detects this energy and sends back a response that contains the tag’s serial number and possibly additional information.
* In simple RFID systems, the reader’s energy pulse functions like an on-off switch. In more sophisticated systems, the reader’s radio-frequency signal can contain commands to the tag, instructions to read or write tag memory, and even passwords.”
* RFID TAGS**:**
* They consist of a tiny processor, memory, and antenna. RFID tags can be read and written wirelessly. They vary in complexity and cost based upon frequency, memory, sensor integration, size, antenna, and power.
* **PASSIVE TAGS:**
* Less complex because reader provides the operating power. They can be packaged small, light, inexpensive (also called Smart Labels) and can last over 10 years.
* But…Range of transmission is relatively short and   
  can require higher-powered readers.

***CHAPTER 3***

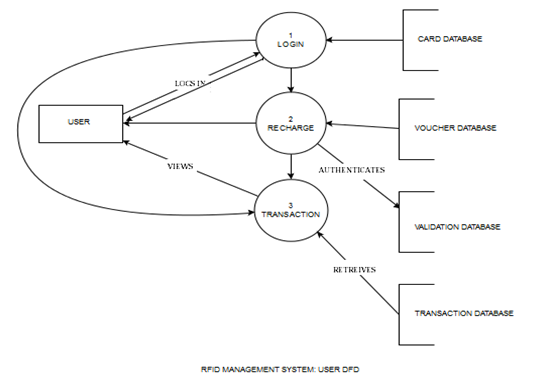
**DESIGN**

**3.1 USE CASE DIAGRAM:**

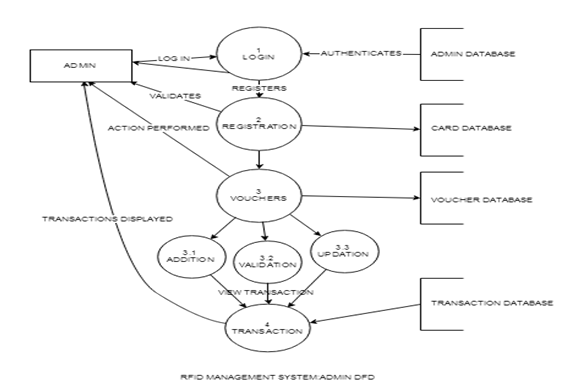


**3.2DATA FLOW DIAGRAM:**

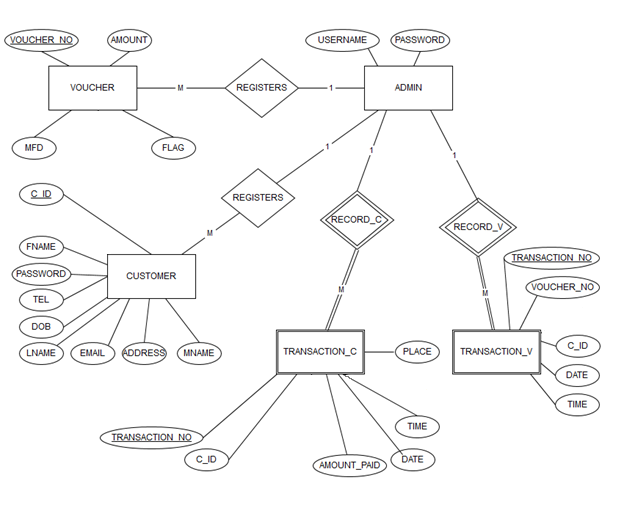
**USER:**

****

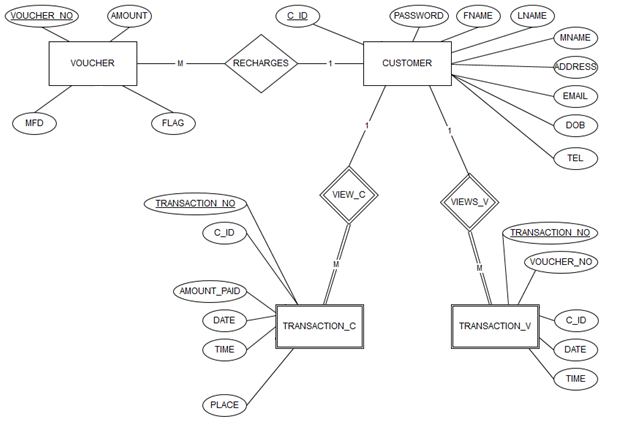
**ADMIN:**

****

**3.3 ENTITY RELATIONSHIP DIAGRAM:**

****

**ENTITY RELATIONSHIP DIAGRAM: ADMIN**

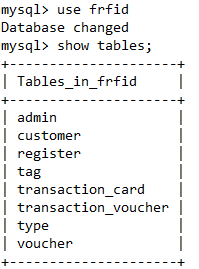
****

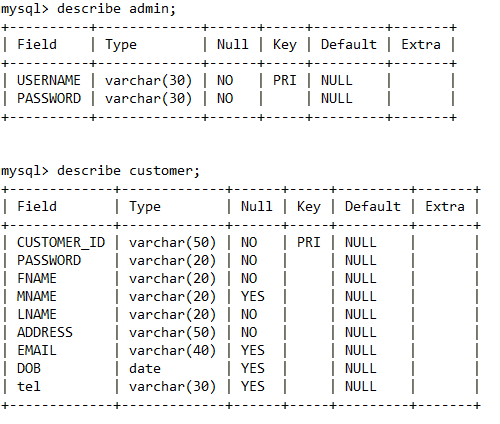
**ENTITY RELATIONSHIP DIAGRAM: USER**

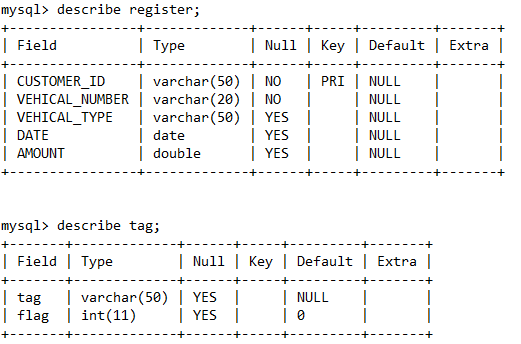
***CHAPTER 4***

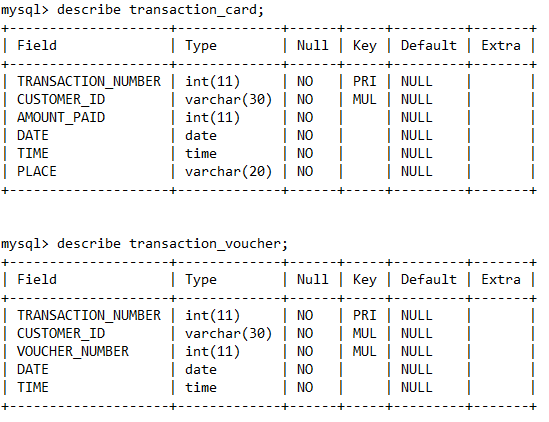
**IMPLEMENTATION**

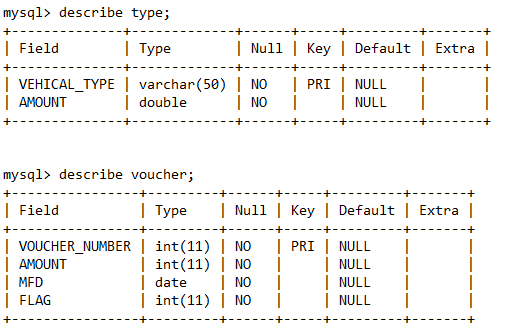
**4.1 TABLE DESIGN:**

****

****

****

****

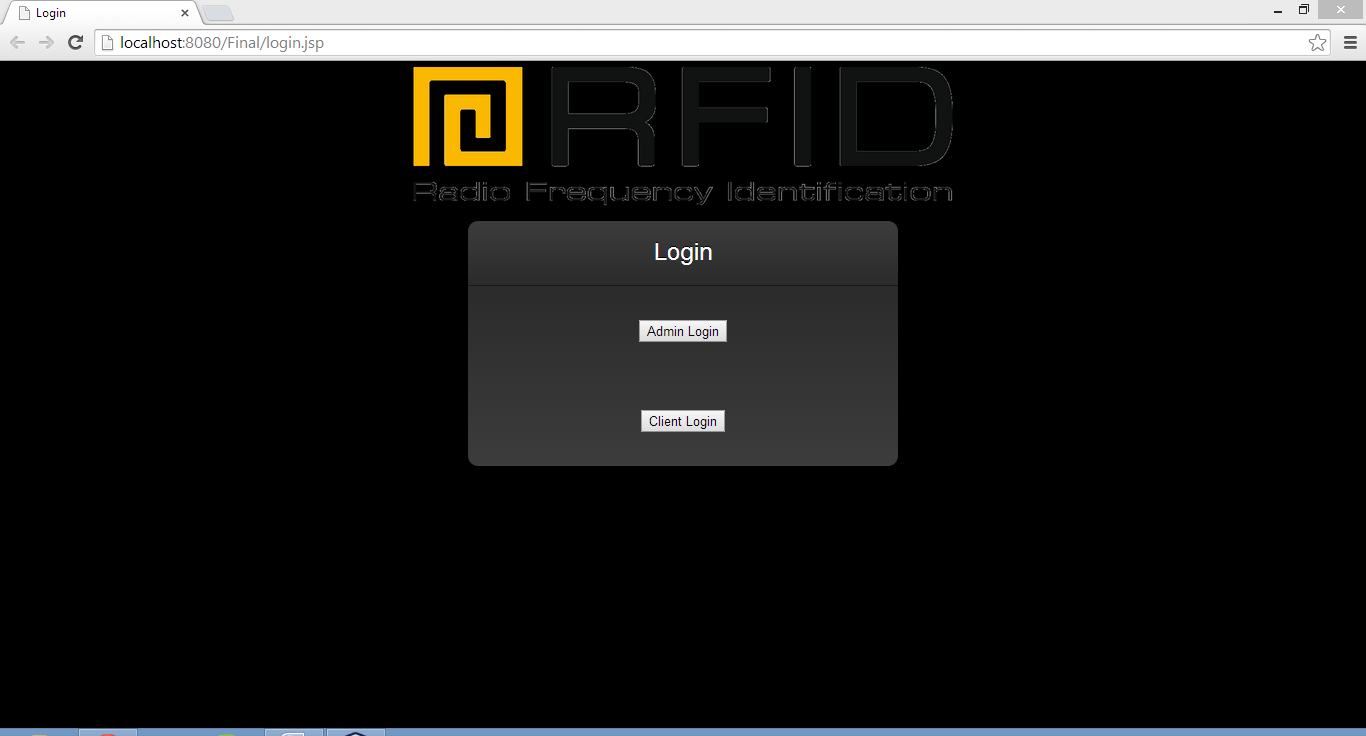
****

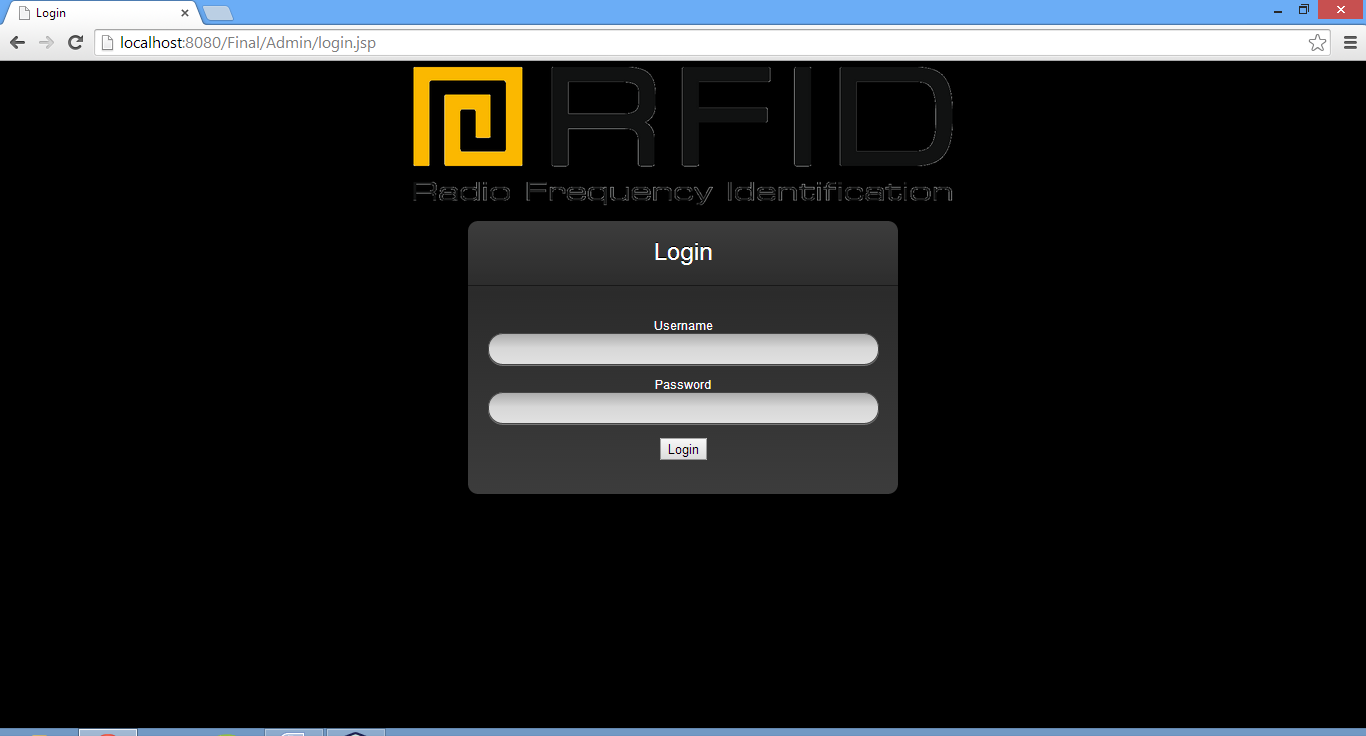
**4.2 FORM DESIGNS:**

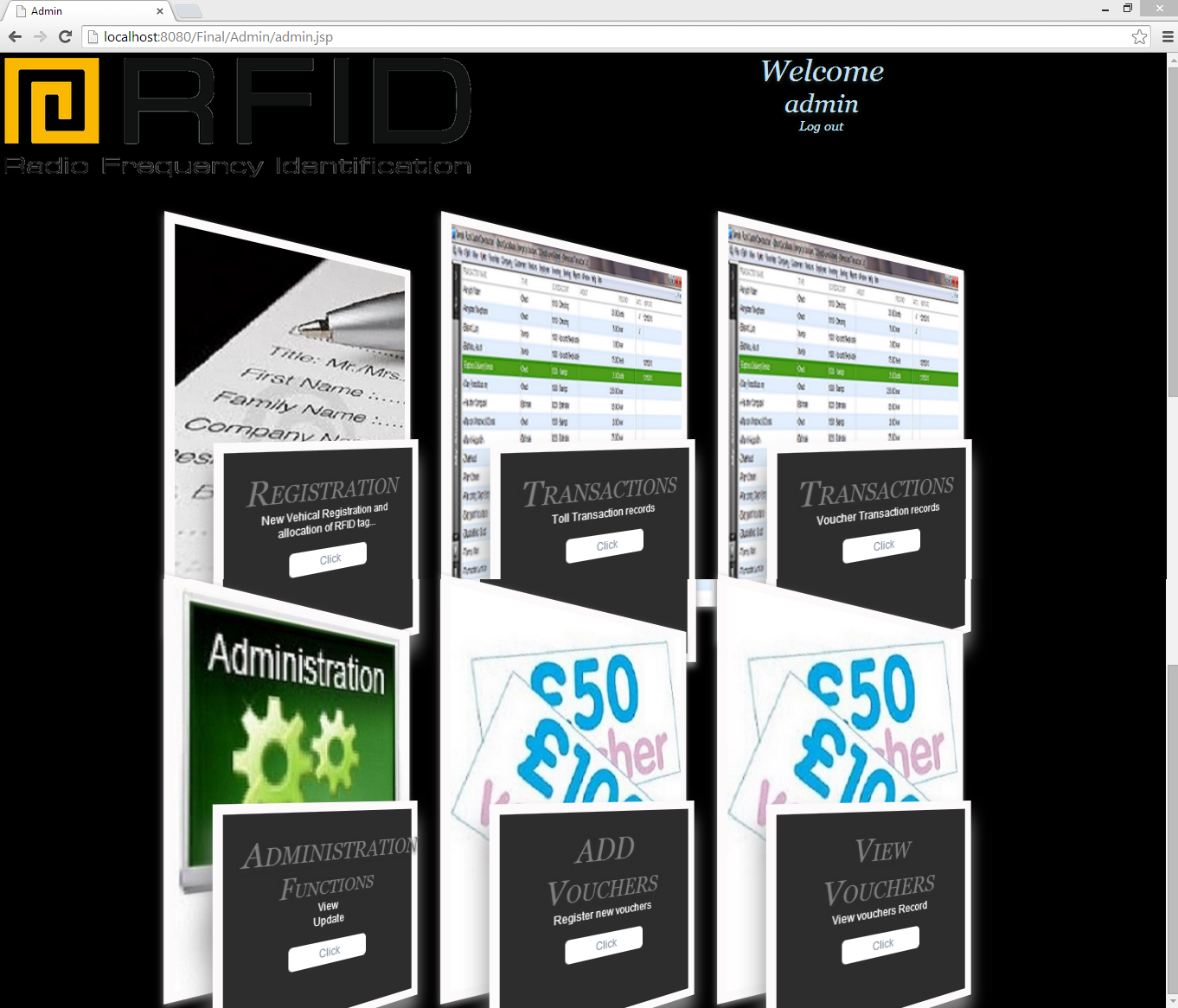
* ADMIN LOGIN**:**

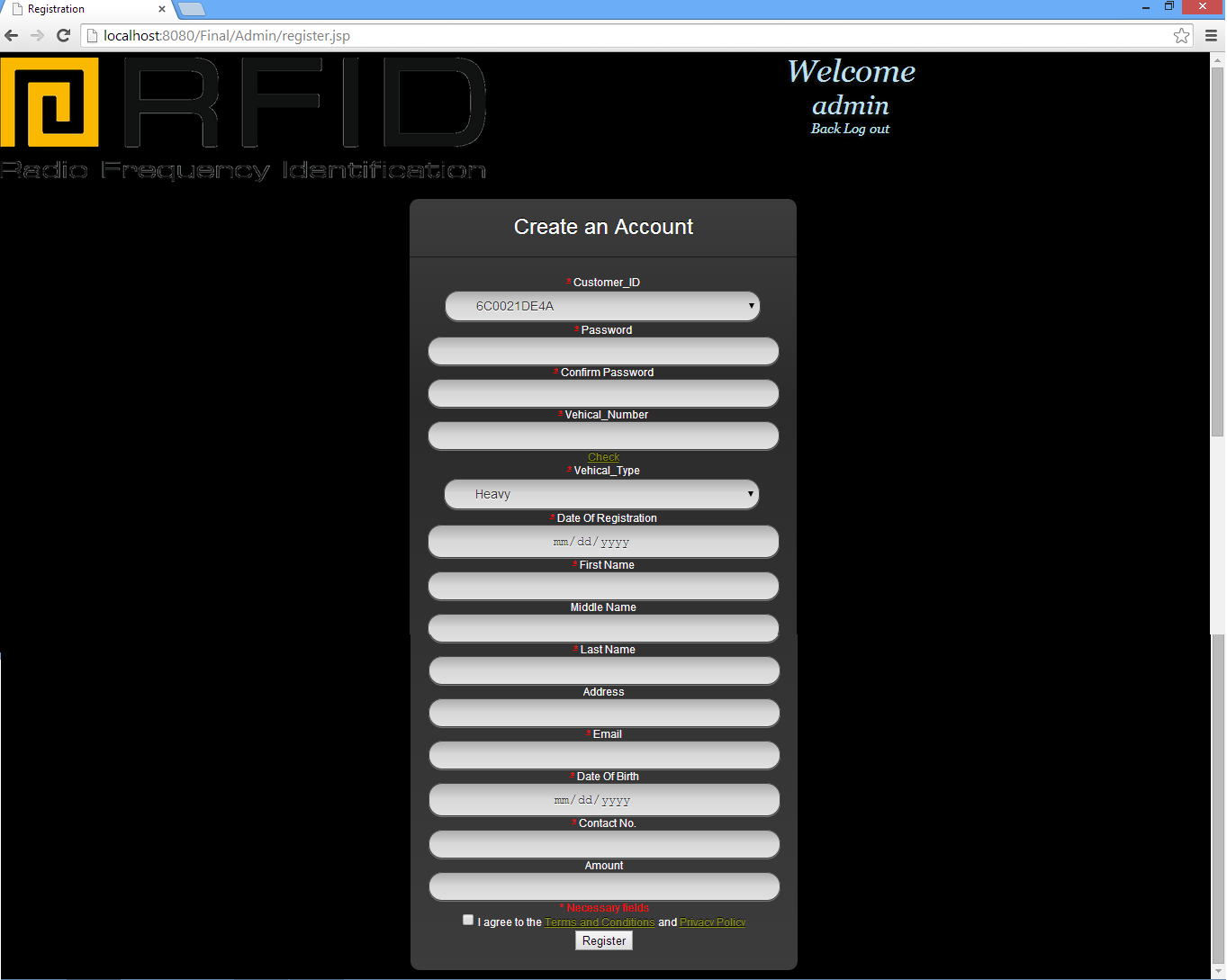
Username: admin

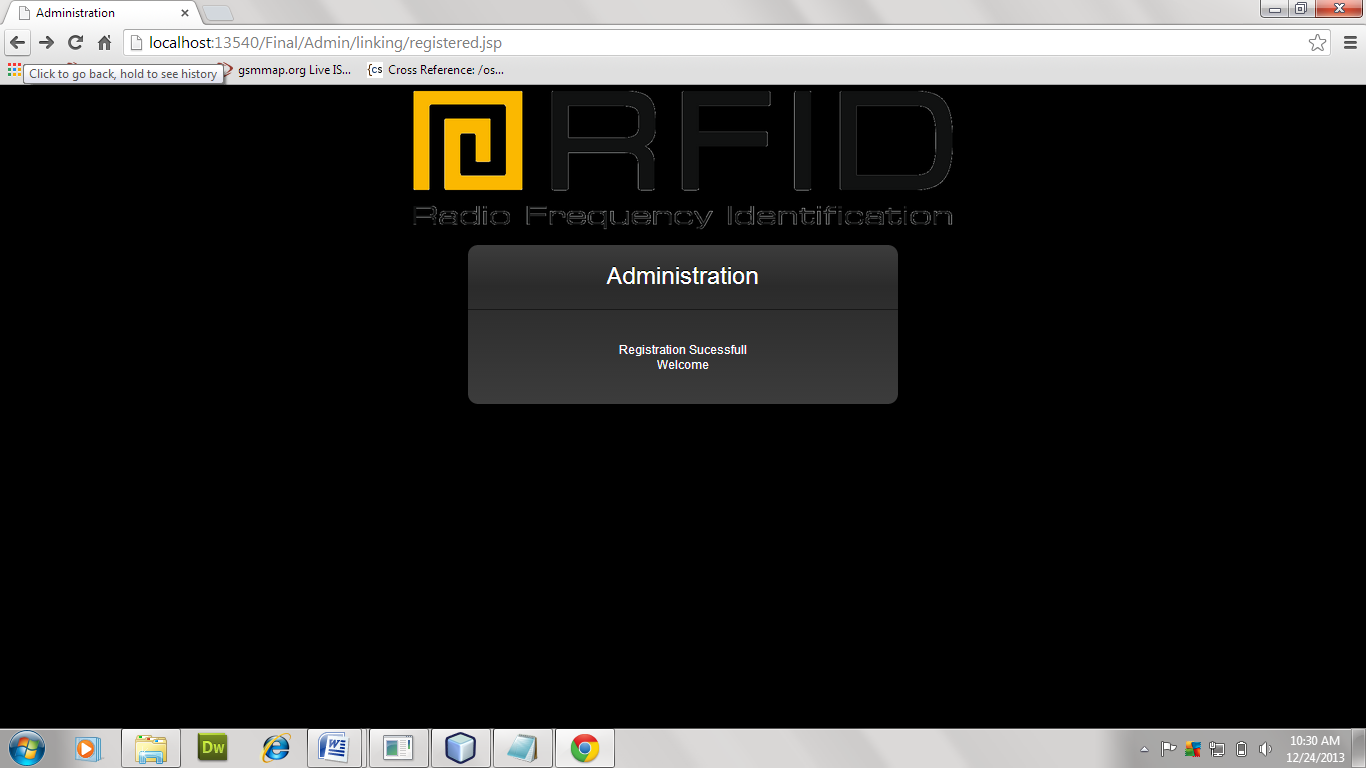
Password: admin

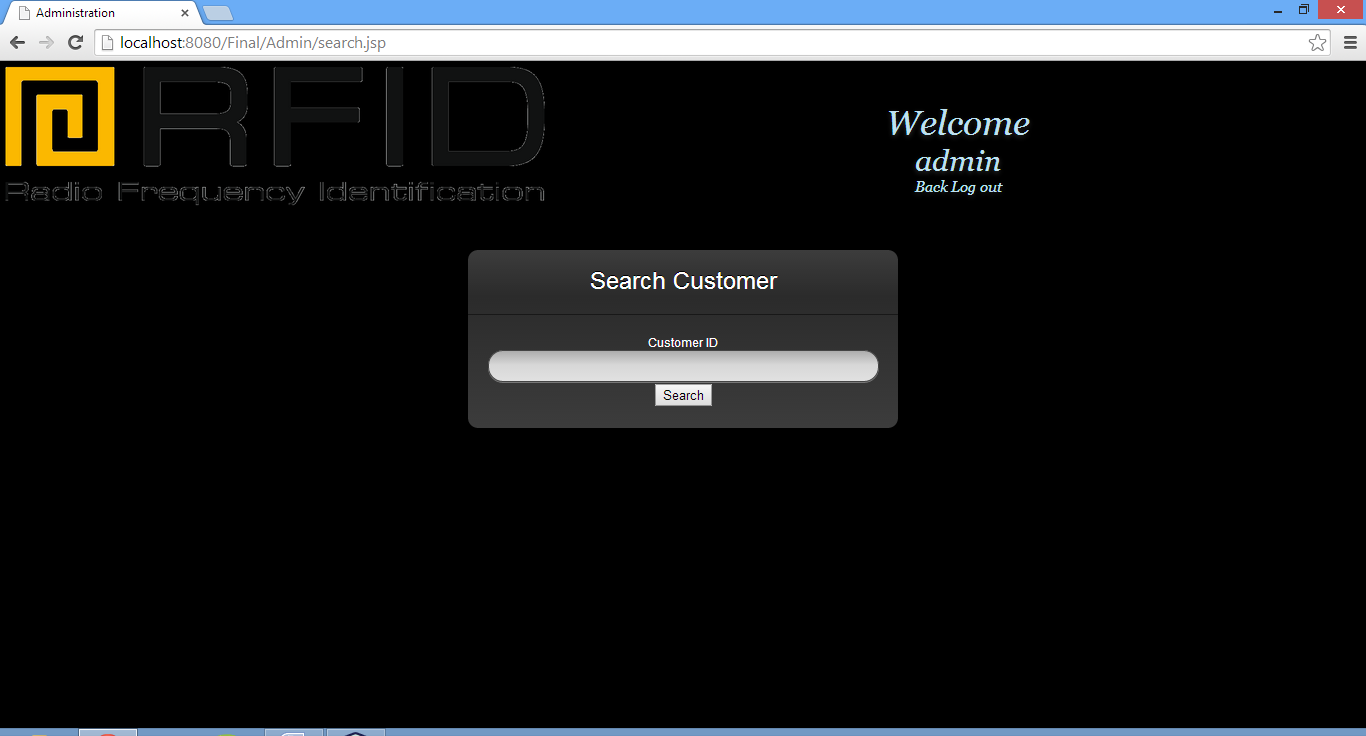
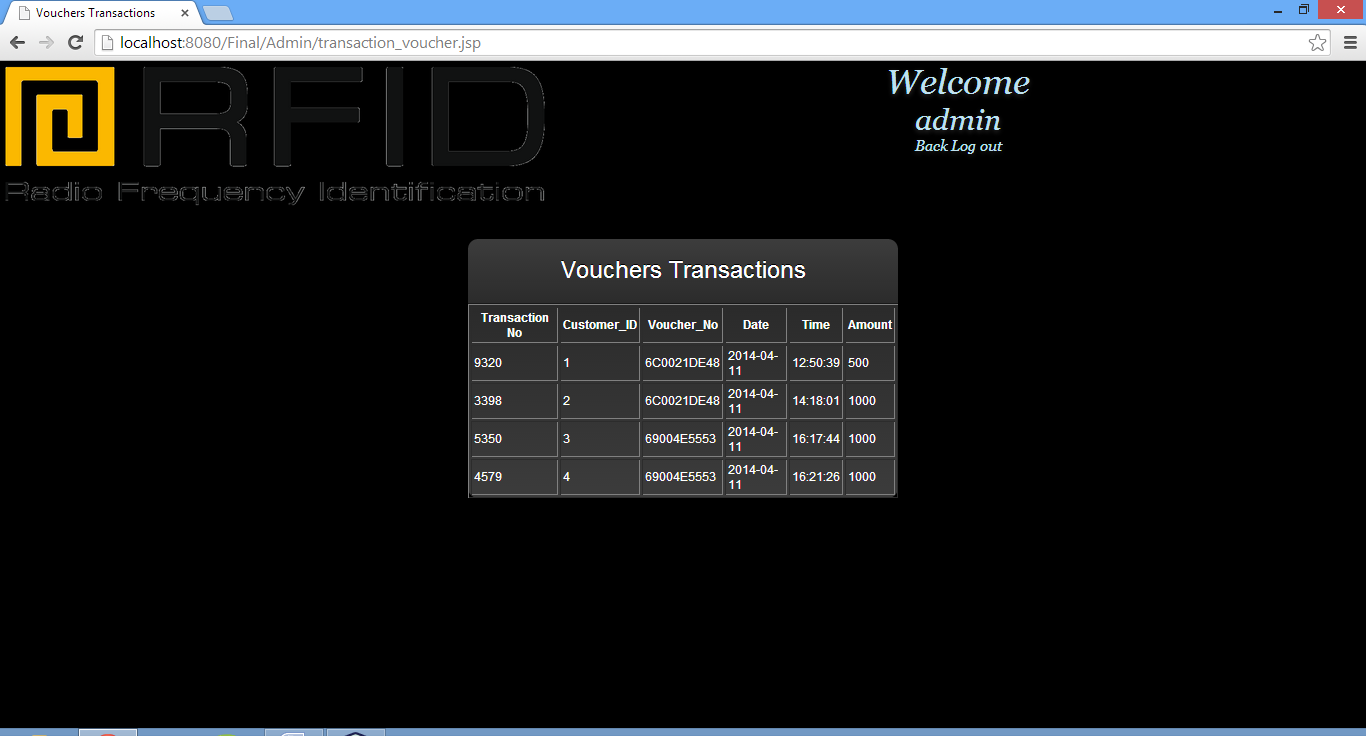
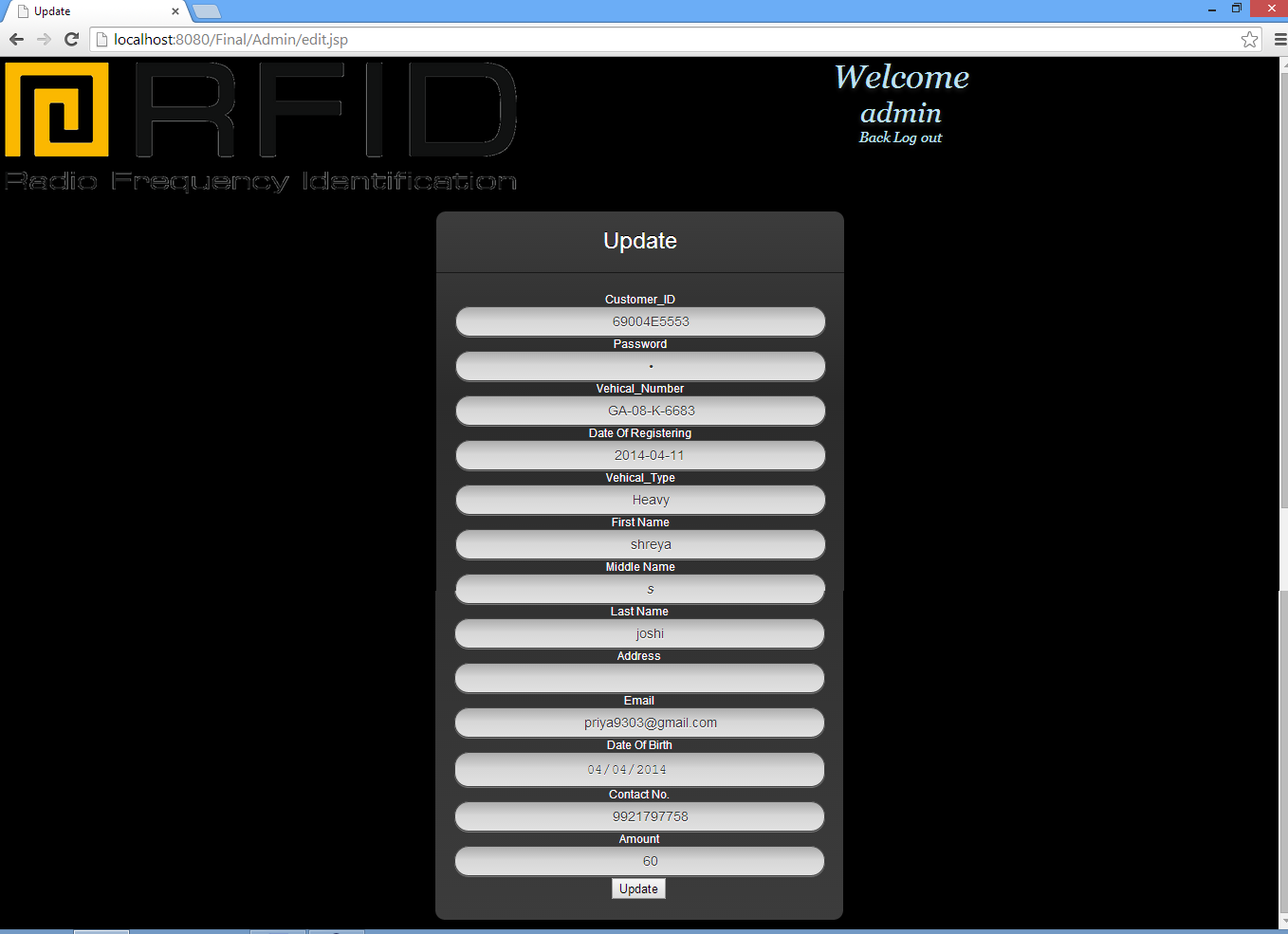
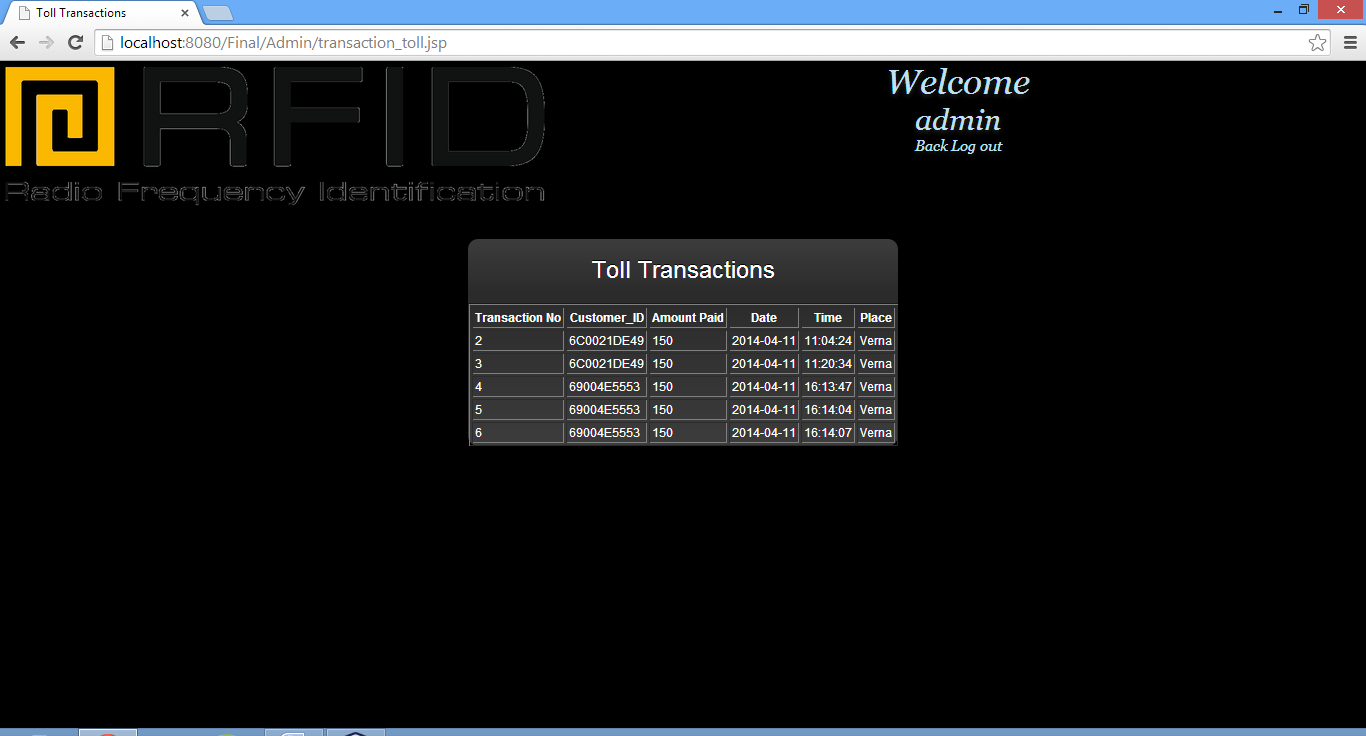




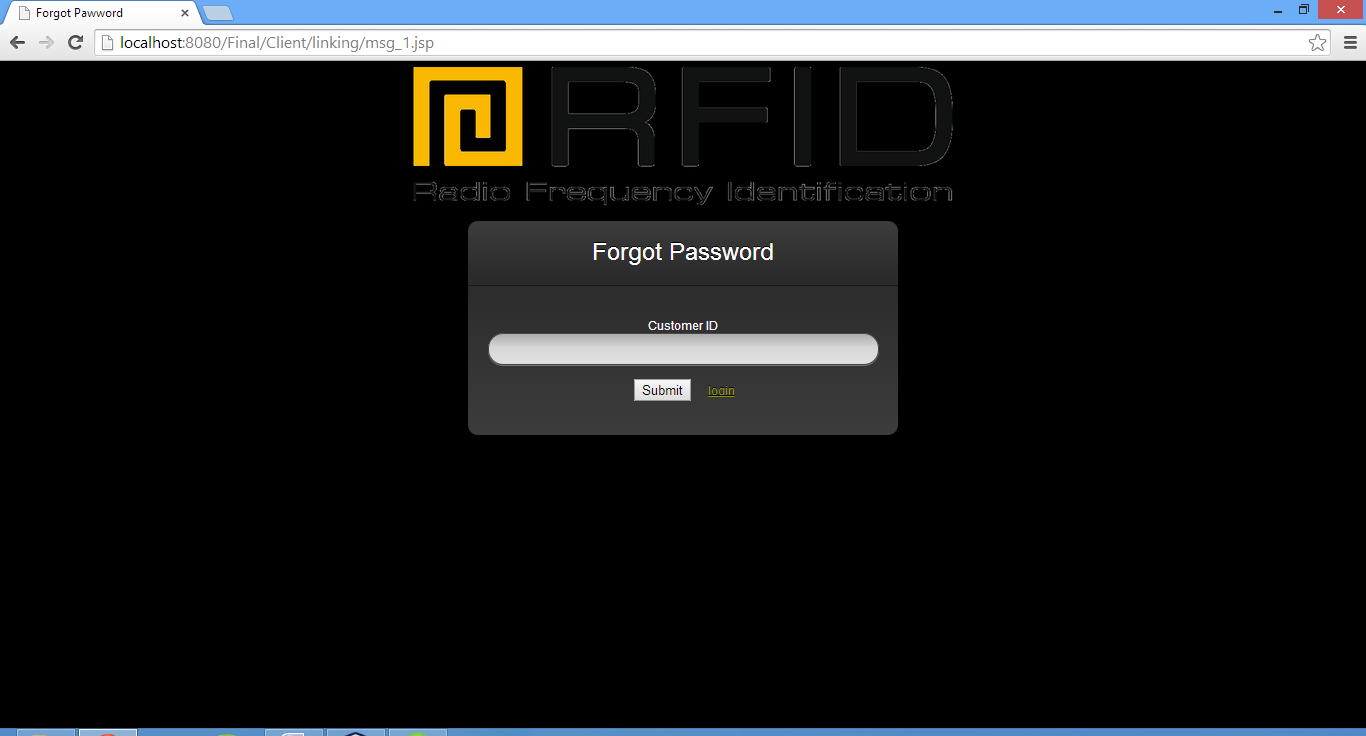
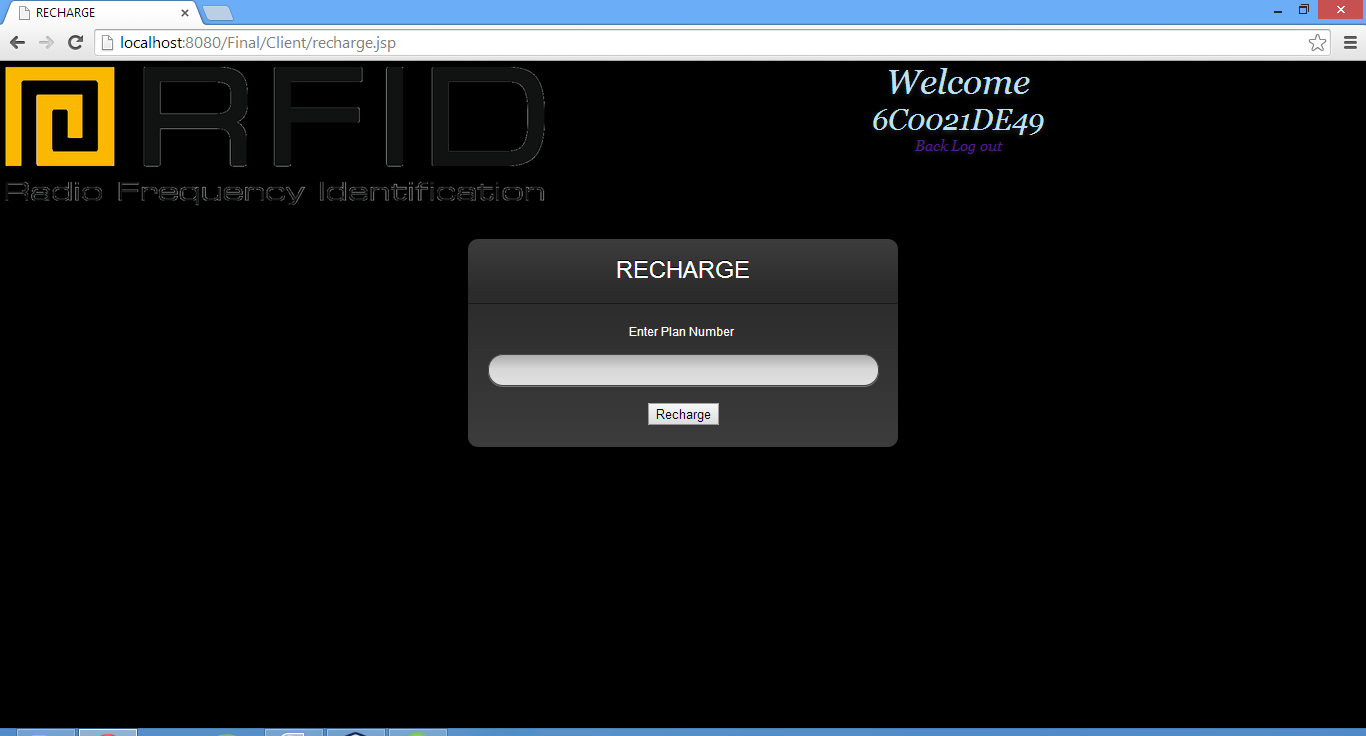
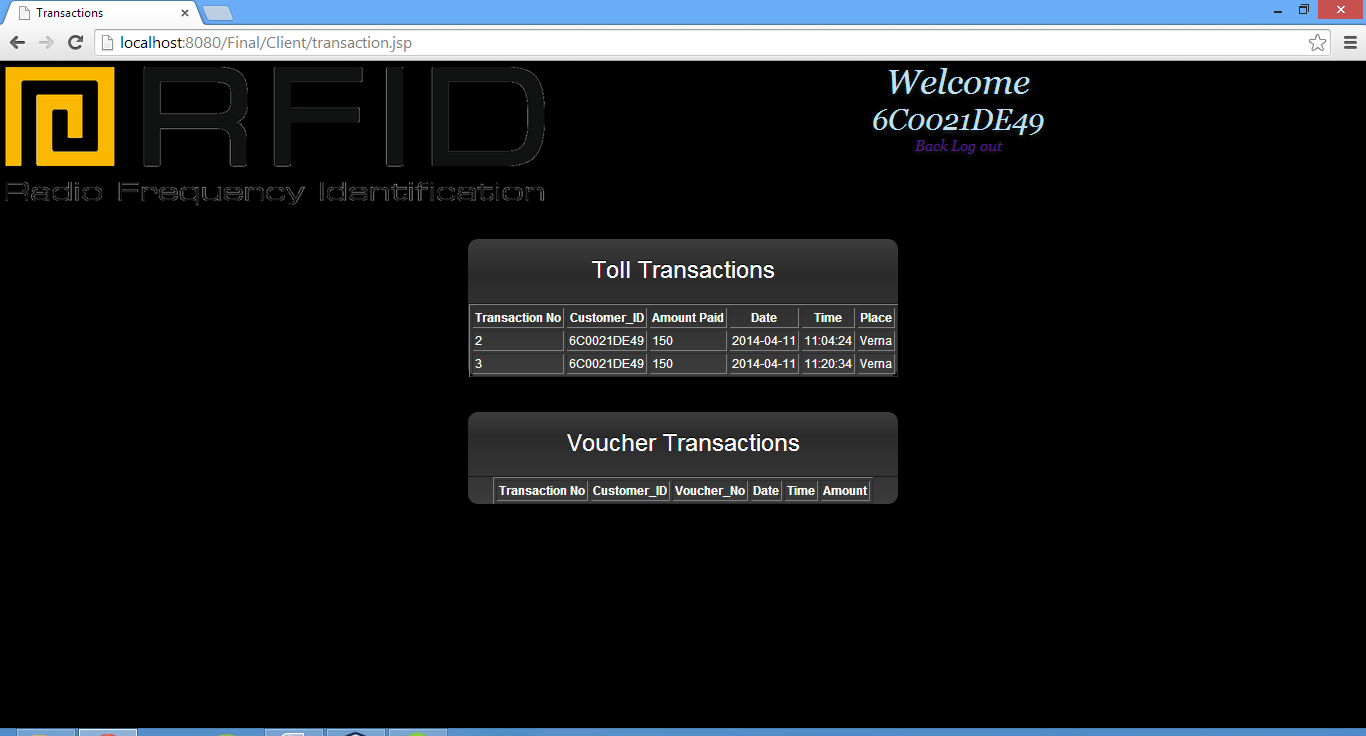


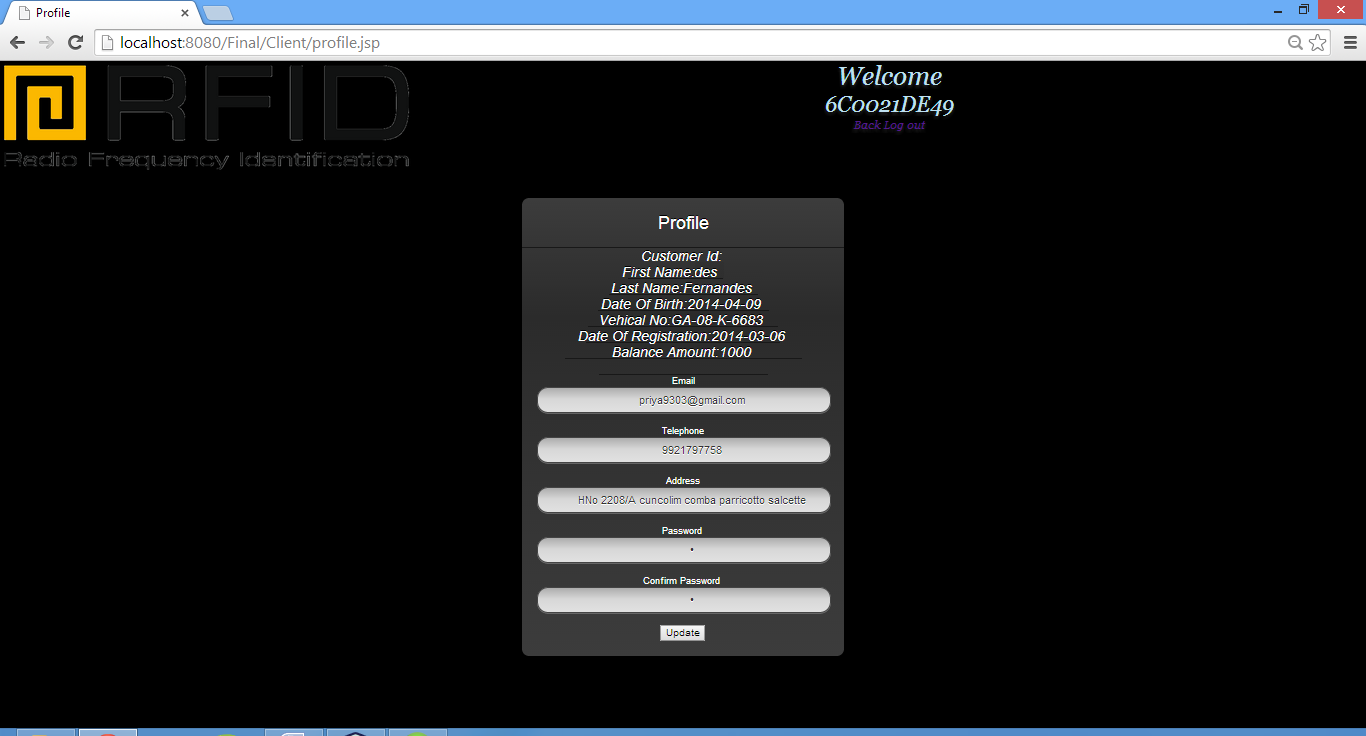
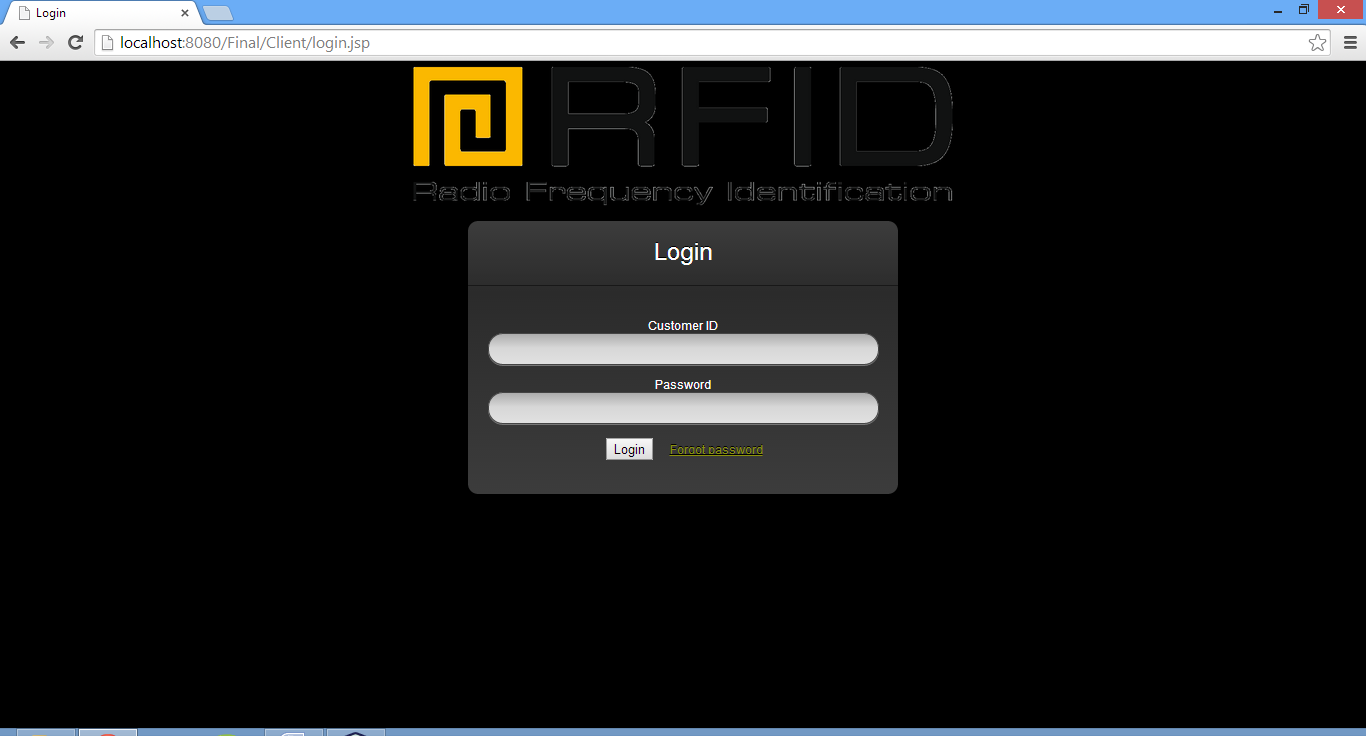




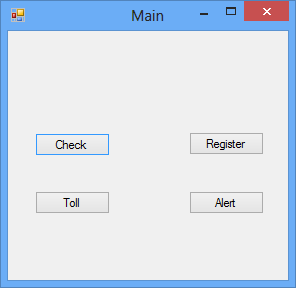


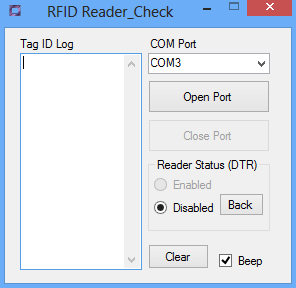
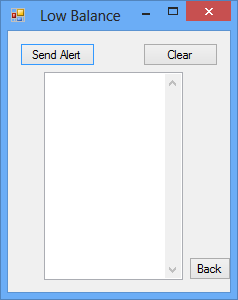
* USER/CUSTOMER LOGIN**:**





* **Toll Deduction Application:**

****

****

***CHAPTER 5***

**TESTING**

**TESTING**

Software testing is the process of evaluating a software item to detect differences between the given input and the expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

* Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

* Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

**5.1 TEST CASES**

In software engineering, a **test case** is a set of conditions or variables under which a tester will determine if a requirement upon an application is partially or fully satisfied. It may take many test cases to determine that a requirement is fully satisfied. In order to fully test that all the requirements of an application are met, there must be at least one test case for each requirement unless a requirement has sub requirements. In that situation, each sub requirement must have at least one test case.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **RFID BASED TOLL COLLECTION SYSTEM** | | | | | | |
| **TEST CASE NUMBER** | **TEST CASE VERSION** | **TEST DISCRIPTION** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **RESULT OF TEST** |
| 1 | 1.1 | User Validation | Wrong ID | Invalid Username Or Password | Invalid Username Or Password | Pass |
|  | 1.2 | User Validation | Wrong Password | Invalid Username Or Password | Invalid Username Or Password | Pass |
|  | 1.3 | User Validation | Correct Password | Login Successful | Login Successful | Pass |
| 2 | 2.1 | Mandatory Fields | No Input | Enter The Required Details | Enter The Required Details | Pass |
| 3 | 3.1 | User Registration | Submit Details | Registration Successful | Registration Successful | Pass |
| 4 | 4.1 | Vehicle Number  Verification | Invalid Vehicle Number | Vehicle Number Not  Registered | Vehicle Number Not  Registered | Pass |
| 5 | 5.1 | Balance | Balance Less Than Minimum Amount | Balance low. Please Recharge. | Balance low. Please Recharge. | Pass |
| 6 | 6.1 | Voucher Validation | Invalid Voucher | Voucher Not Found | Voucher Not Found | Pass |
|  | 6.2 | Voucher Validation | Used Voucher | Voucher Already Used | Voucher Already Used | Pass |
|  | 6.3 | Voucher Validation | New Voucher | Recharge Successful | Recharge Successful | Pass |

**FUTURE SCOPE**

* To ease the mode of payment - transactions and recharge of vouchers, net banking can be integrated as a part of the project.
* The project can also be extended to possessing greater automated location tracking capability.
* The project can be secured against impersonation (exchange of tags) by incorporating decision recognition application that serve to increase security at the booth

**BIBLIOGRAPHY**

* Klaus Finkenzeller, "RFID Handbook: fundamentals and applications in contactless smart cards and identification," Wiley press, 2003.
* Michael, K., McCathie, L., "The Pros and Cons of RFID in Supply Chain Management," International Conference on Mobile Business, July 2005.
* Georgian Electronic Scientific Journal: Computer Science and Telecommunications 2010|No.1(24)