Software Engineering Experience- Fraud Detection

Suhair Amer and Zirou Qiu
Department of Computer Science,
Southeast Missouri State University, One University plaza, Cape Girardeau, MO, USA 63701

Abstract – an undergraduate student produced documentation and implemented a preliminary system as part of the requirements of a software engineering course. The developed system is concerned with credit card fraud detection system. G score was the key element that the system used to determine if the transaction is a fraud action.

Keywords: *G score, fraud detection, software engineering.*

1 Introduction

Credit card fraud is defined as:" unauthorized use of a credit/debit card, or card number, to fraudulently obtain money or property" (Brody et al, 2014), which is an increasing issue in every country. Around ten percent of Americans were victims of credit card fraud and $5.55 billion in credit card fraud was reported in 2012 worldwide. (Brody et al, 2014). In terms of number of people who own a credit card, in 2012, 60% of college seniors owned a credit card, 27% of people over the age of 50 had four or more credit cards (Arora et al. 2015). Because of the huge number of credit card users and use of technology, having a functional and effective credit card fraud detection system has become necessary for everyone. Currently, there are several ways used by big companies to prevent illegal actions from happening, such as, Card Verification Value Plus CVVplus. On the back of almost every major credit card there is a Card Verification Value CVV to verify payments when people do shopping online. But that CVV code is static and is not considered private once someone else knows it. The CVVPlus system provides consumers with a code that is updated daily (Kerner 2015). This is a good idea when it comes to fraud detection system design. Some scientists also worked on developing a new binary support vector system to increase detection rate of card fraud. This involves artificial intelligence design to learn the spending pattern of users (Chen and Chen, 2006). This design is very classic and could effetely detect suspicious activities. Currently in U.S., the most common type of card fraud is card-present fraud, where a user physically swipes the card using the card reader. EMV chip technology, which embedded a smart chip in the card, provide an extra shield of protection to a credit card (Marshall and Lattin, 2015). Major banks are now updating their cards with this new technology. There are many other ways to help user detect unrecognized behavior, such as, distributed anomaly detection using cooperative learners and association rule analysis (Deshmeh and Rahmati, 2008).

The student in the Software Engineering course was required to use a software process to complete documentation related to analysis, design, object design, implementing and testing. To relate these steps to a real time project, the student chose to develop a credit card fraud detection system that uses user spending patterns and location to check for unusual patterns. The system stores previous transaction patterns for each user. Based on the user's spending ability and country, it calculates user's characteristics. If more than 20 -30 % deviation in user's transaction (spending history and operating country) is detected, it will be considered an invalid attempt and the system takes action (require the user to login again or even block the user for more than 3 invalid attempts).

2 Analysis

The purpose of the fraud detection system is to protect user's data and increase the security level of credit cards. It learns users' spending patterns, configure users' information and automatically notifies users or locks the account when unidentified actions happen.

The target customers are banks and their clients who have credit cards. The number of users we are expecting is 1000. The geographic region the system is going to cover is our county.

The objective and success-criteria of the project is that the system will formulize the user's dynamic spending pattern based on first N purchases, comparing each purchases with the pattern and update the pattern gradually. Many factors are considered while building the pattern database which include: user's approximate

annual income; gender; address; when and where most of the transaction had taken place; the average amount per transaction; the frequency of using the ATM; user's tipping habits; categories of items that had been bought and so on. In terms of comparing the most recent purchase P1 with the pattern P, system will analysis a set of figures G, which includes, the distance between P1 and P, the amount of transaction between P1 and P, the time (i.e. attempting to use the card at 2 am when the user never used it after 1 pm before) of P1 and P and many other factors, to come up with a numeric value Gs. Based on the value of Gs, system will determine the suspicious level S of P1 from grade of 1(least) to 5(most) and make corresponding actions (whether it should let it go through without notifying users, let it go through with notifying users, or block the transaction while it is pending and lock the card).

The goal of this project is to design credit card fraud detection system which could potentially learn users' spending patterns, configure users information and automatically notify users or lock the account when unidentified actions happens depending on a set of figures G.

The Functional Requirements are:

1. The system will formulize the user's dynamic spending pattern based on first N purchases, comparing each purchases with the pattern and update the pattern gradually.

2. Many factors are considered in building the pattern including user's approximate annual income; gender; address; when and where most of the transaction had taken place; the average amount per transaction; the frequency of using ATM; user's tipping habits; categories of items that had been bought and so on.

3. In terms of comparing the most recent purchase P1 with the pattern P, the system will analyse a set of figures G, which includes the distance between P1 and P, the amount of transaction between P1 and P, the time of P1 and P and many other factors, used to come up with a numeric value Gs.

4. Based on the value of Gs, the system will determine the suspicious level S of P1 from grade of 1(least) to 5(most) and make corresponding actions (Whether it should let it go through without notifying users, let it go through with notifying users, or block the transaction while it is pending and lock the card).

Table 1 explains what happens when some one uses a card.

Table1: scenario regarding checking who used the card

***Scenarios name*:**	WhoUsedMyCard
Participating actor instance:	Alex: card owner Ben: bad guy
Flow of events	1. Ben, who hacked into Alex's computer and got his card information at 1am and bought many food online. 2. After comparing the user's spending pattern, the system found that the location where transaction happened and the time doesn't match and the suspicious level is really high. The system immediately lock Alex's account and send a message to Alex, notifying this unrecognized action. 3. Alex receives the acknowledgment, after realizing the card has been used by someone else, he called the card company and do the further process.

For the use case model, the following were identified:

- ***Client*:** The card owner
- ***Card***: the credit card issued by the card company
- ***Previous record***: The previous fraud record of this card
- ***No fraud action***: the transaction suspicious level is 1, no action should be made
- ***Suspicious level 2***: the system change the suspicious class of the transaction to level 2
- ***Suspicious level 3*:** the system change the suspicious class of the transaction to level 3
- ***Suspicious level 4*:** the system change the suspicious class of the transaction to level 4
- ***Suspicious level 5***: the system change the suspicious class of the transaction to level 5
- ***System manager***: the system manager gets noticed if the transaction level is higher than 2

3 Design

The goal of this project is to design credit card fraud detection system which could potentially learn users' spending patterns, configure user's information and automatically notify users or lock the account when unidentified actions happens depending on a set of figures G.

Table 2, lists an example of an entity-objects table. Table 3, lists an example of a boundary-objects table.

Table 2 : Entity-objects table for ReportUndetected

Entity Object	Attributes & Associations	Definition
Account	1. balance 2. history of fraud action 3. credit history 4. history of transaction	A bank account owned by clients who are currently using our fraud detection system.
customer	1. name 2. account 3. address 4. type	Customs are clients who own credit cards and use our fraud detection system.
Card reader	1. sequence number 2. Company	Card reader is the machine which user uses to make transaction happen.

Table 3: Boundary objects table for ReportUndetected

Boundary Object	Definition
Login form	The interface where user can log in with their account and password
Report Submit Button	The button used by a customer to initialize the reportundetected use case.
Report form	The interface where user can report a suspicious transaction which the system didn't detect
Report reply	The interface which system replies to user's report
Workstation	The station system uses to analysis user data

An example of a Control objects is Report-Undetected-Control which manages the reporting form and reporting reply function. This object is created when report form is filled and submitted. Then it collects the data from the account and the form that is analyzed by the workstation and then send to the user through report reply. Also, it will record the sequence number of the card reader the user used.

With regard to persistent data management, the persistent data include a user's password, history of transactions, card information, etc. They will all be stored in MySQL database.

With regard to access control and security, code was added to prevent MySQL injection and JavaScript injection. All the passwords in the database are encrypted by PHP password hashing. Specific users have access to a particular database and system only grants them the least privileges.

With regard to boundary conditions and the Start-Up condition, it generally takes 5 seconds for the system to start up. It generally takes 10 seconds for the system to shut down completely.

Different design patterns have been checked to identify what works and what does not work with the system.

- Pattern 1: Bridge design pattern would work with the system because the system need to decouple the interface of user class from its implementation and the developer should not be constrained by the existing component.
- Pattern 2: Adapter design pattern would work with the system because some PHP code need to be encapsulated since they were not designed to work with the system.
- Pattern 3: Strategy design pattern would work with the system because so many algorithms need to be decoupled from their implementations. For example, the hash function used MD5.
- Pattern 4: Abstract Factory design pattern would not work with the system because the system doesn't need to encapsulate the creation of families of related objects since are no related objects.
- Pattern 5: Command design pattern would not work because the system doesn't have any object responsible for command processing.
- Pattern 6: Composite design pattern would not work because there is not hierarchies nor superclass in the system.

4 Implementation and results

The system was implemented as a client – server model, using HTML, JavaScript and PHP. For the client component, HTML and JavaScript are used for GUI and client side validation. PHP is used to perform server data validation, establish connection to the database, store and retrieve data from database, and calculate G score.

Forms are implemented with HTML. After clicking each input field, a corresponding JavaScript function is called to check if the data is valid. After submitting data through post method, PHP will first check again if the data is valid to protect against web proxy attack. Then if valid, it will connect to the database and store the data. After logging in, user's transaction history will be retrieved and outputted to the browser.

User can successfully register, log in, process transactions, view history and detect fraud actions.

In the main page [Figure 1] the user is required to enter his/her email address, provide his/her password and then click on the "Log in" button. The users have the option to retrieve their passwords if they forgot them by clicking on the "Forgot Password?" link. If this is a first time user, he/she can register by clicking on the "Sign Up" link.

If this is a first time user, he/she will click on the "Sign Up" link which will direct the user to a new form, [Figure 2] which will ask for information such as name, birthday date, password, security question, etc.

The system requires the user to enter the password in a specific format. If this format is not met, the system will produce an error and asks the user to enter a new password [Figure 3]. The entry box of the password will be colored red if there is an error.

The system also checks and produced an error if both provided passwords do not match [Figure 4]. The entry box of the password will be colored red if there is an error and colored green if it is correct.

Once both passwords match and are in the correct format, the password is accepted and the box is colored green [Figure 5].

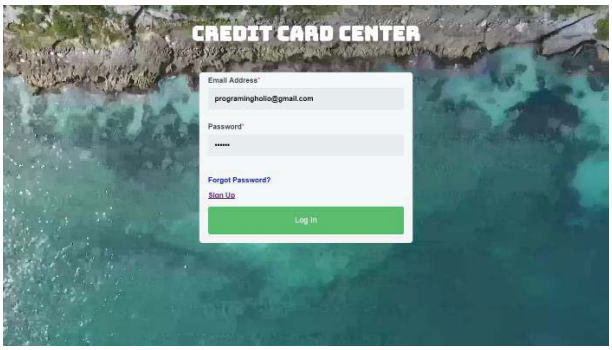

Figure 1: main page

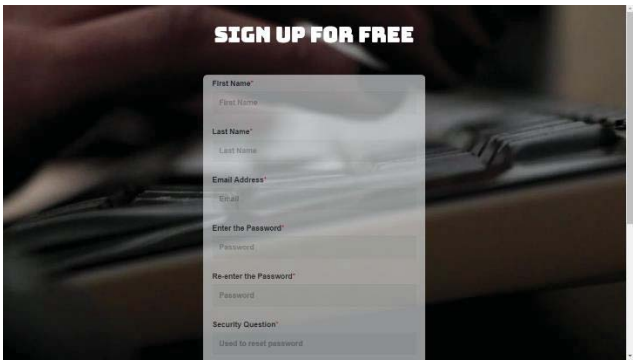

Figure 2: Sign up page

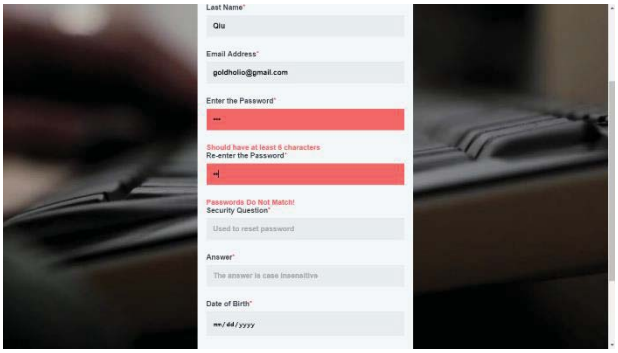

Figure 3: Error when password is in not in correct format (in red)

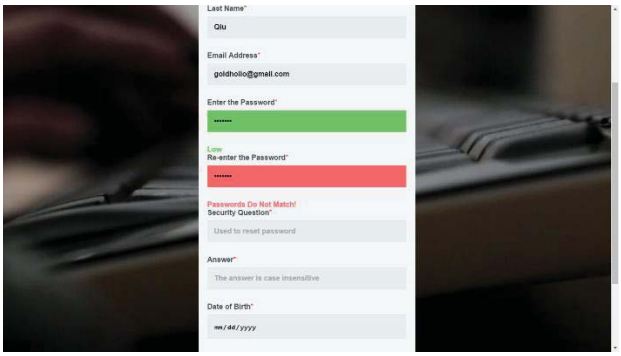

Figure 4: Error when re-entered password does not match (in red).

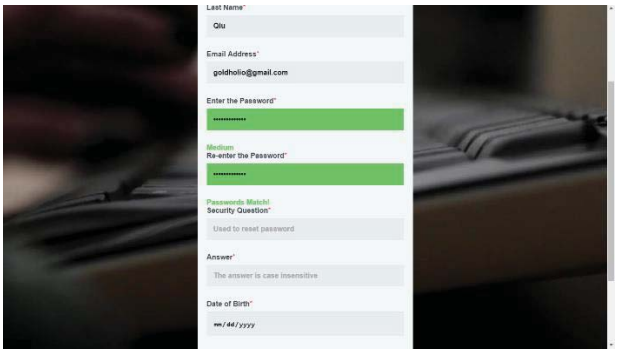

Figure 5: Password re-entry accepted (in green).

Once the user is finished entering all required information [Figure 6] they can click on the "Get Started" button. The user has the option to cancel creating a new account by clicking on "Already have an account" link.

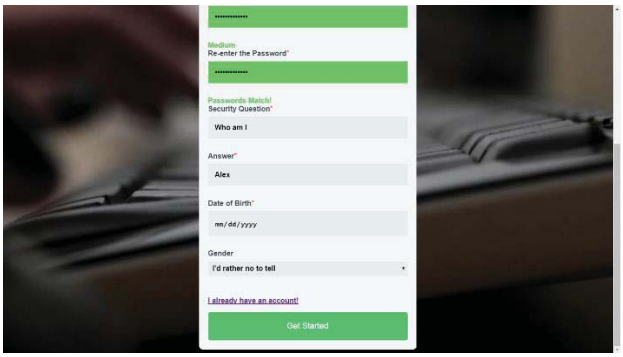

Figure 6: form is completed and ready to click "Get Started"

Once the user has successfully created an account, they are required to log in through the main form, [Figure 1]. The user will enter his/her email address and password, then click "Log in" button. If the email address and password do not match, an error message will be displayed [Figure 7]. The user's account will be blocked after 4 failed attempts.

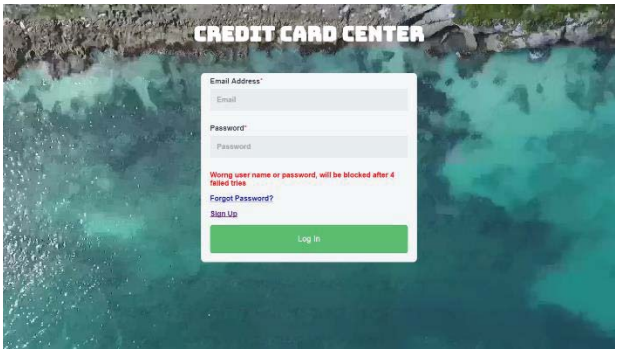

Figure 7: error message after entering the wrong user name or password.

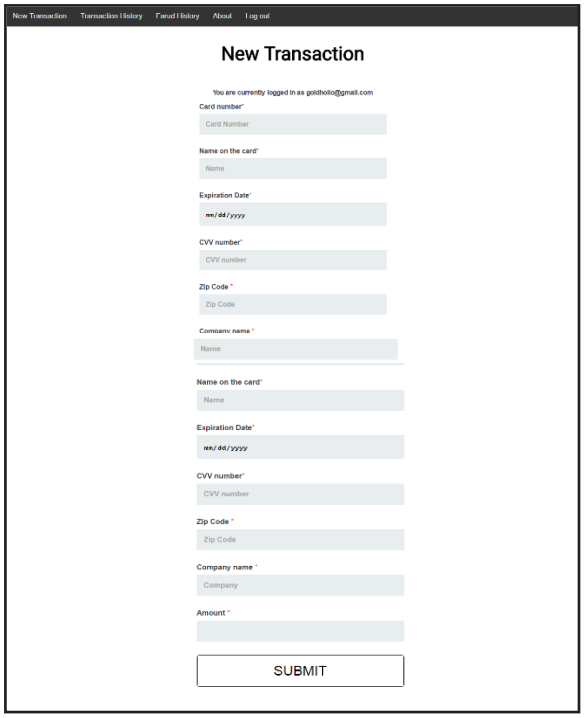

Figure 8: want to start a new transaction. Chose "New Transaction" option.

Once the user accesses the system, he/she can choose one of the following items appearing on the top part of the New Transaction menu [Figure 8]:

- "New Transaction" where a new transaction is occurring and an amount is being withdrawn.
- "Transaction History" which displays a log of all transactions related to this account, [Figure 9].
- "Fraud History" which displays a log of all transactions that are thought to be fraud.
- "About" which describes the project.
- "Logout" to logout of the system or this form.

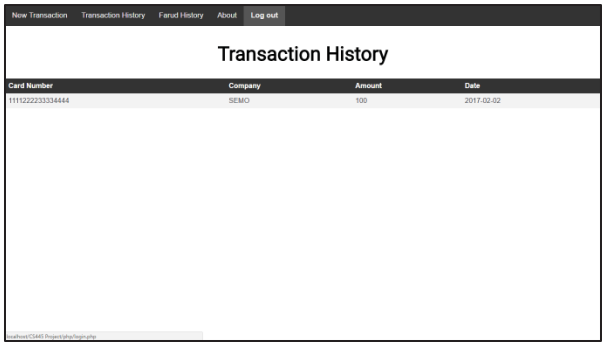

Figure 9: list of transactions.

5 Conclusion

This paper described the process of developing a system as part of the requirements of an undergraduate Software Engineering course. The student was required to apply the concepts studied in the course to develop a system that detect credit card fraud and produce all necessary documents and graphs for the analysis, design, object design, implementation, testing stages.

The main objective of the course is not to develop a system but to apply concepts in developing a simple system to give them an idea about the process. In addition, the student had less than a month to implement and test the system. Therefore, this is not, in any means, a fully functioning system. The student was able to provide a working prototype of the interface that would be used if a user would use his/her card and receive feedback on whether this was actually the user or a fraud action. G score was the key element that the system used to determine if the transaction is a fraud action.

The student successfully completed the requirements of the course and after doing this project, the student indicated that he had a deeper understanding of PHP and web security.

6 References

Arora, R., Gupta, D., & Pahwa, P. (2015). Fraud Detection Life Cycle Model: A Systematic Fuzzy Approach to Fraud Management. *International Journal of Computational Intelligence & Applications*, *14*(2).

Berghel, H. (2007). Credit Card Forensics. *Communications Of The ACM*, *50*(12), 11-14.

Brody, R. G., Brown, D. M., Chettry, A., & White, W. I. (2014). Proliferation of Credit Card Fraud with Current Technological Advances. *Insights to A Changing World Journal*, 2014(2), 92-107.

Chen, R., Chen, T., and Lin C. (2006). A new binary support vector system for increasing detection rate of credit card fraud. *International Journal of Pattern Recognition & Artificial Intelligence*, 20(2).

Davis, K. (1998). The Bonnie and Clyde of credit card fraud. (Cover story). *Kiplinger's Personal Finance Magazine*, *52*(7), 65.

Deshmeh, G., & Rahmati, M. (2008). Distributed anomaly detection, using cooperative learners and association rule analysis. *Intelligent Data Analysis*, *12*(4), 339-357.

Kerner, S. M. (2015). Tender Armor Adding New Layer of Security to Credit Card Transactions. *Eweek*,

Marshall, E. A., & Lattin, M. (2015). Understanding the Payment Card Fraud Liability Shift. *Commercial & Business Litigation*, *17*(1), 15-18.

Porkess, R., & Mason, S. (2012). Looking at debit and credit card fraud. *Teaching Statistics*, 34(3), 87-91.