AI Banking Agent

Nikhila Kamat[1]
Computer Engineering
Xavier Institute of Engineering
Mumbai, India
nikhila300@gmail.com[1]

Michelle D'cruz[2]
Computer Engineering
Xavier Institute of Engineering
Mumbai, India
michelledcruz30o@gmail.com[2]

Carmelite Dabre[3]
Computer Engineering
Xavier Institute of Engineering
Mumbai, India
carmelitadabre0@gmail.com[3]

Abstract—Our application is designed for the web-clients of banks. Although, banks are always looking out for powerful and user friendly applications while upgrading their website [18], our platform-independent AI application uses an artificial algorithm -"String Matching" for analyzing and understanding the queries of the users like bank's loan, account, policy etc. No specific format would be required for the users to ask questions, for instance, the question – "May I know about the personal loan tenure?" can be also asked as – "What is the personal loan tenure?" In additional to it, a user friendly speech-to-text module has been incorporated to help senior / disabled, where animated talking character would read out the answers before displaying them on the screen, and it also features ATM finder and Branch locator modules to guide the web-clients with the locations.

Keywords-AI, Banks, String Matching, user friendly

I. INTRODUCTION

Understanding the etymology of title "AI BANKING AGENT" –

AI - Our application uses an artificial algorithm called as "String Matching" and applies its built-in artificial intelligence for analyzing the queries asked by the web-clients and answers them suitably.

BANKING - This application is designed specifically for the benefit of the bank as it will reside on the central server of the bank and can be accessed over a network.

AGENT - It is like a software robot that acts as bank's agent, having perceived its environment, it takes actions which maximize its performance measure and chances of success. Nevertheless, a logical agent solves specific problems and portrays interaction and communication between the web-client and the agent. [10]

II. MOTIVATION

AI, better known as Artificial Intelligence, is a key component identical to the functioning of brain; however, banking related questions need to be dealt with reputation of brand image and its perception. [5] Yet, it should make sense of interactive data of both structured and unstructured in a way that supports the experience, [17] as a result of which, this gets exponentially more difficult as data increases. Thus, AI gets 'smarter' by virtue of more data, but, its complexity increases. [4] Our AI application tries to interpret what web-client wants to ask and immediately provides suitable answers to the queries they ask, related to bank. [20]

As elderly people find it difficult to operate the bank website for the queries, some design was needed to make it user friendly for elderly and this motivated us to design a solution.

III. AIMS AND OBJECTIVES

A. Aims

- Combining various human expert intelligence w.r.t the bank.
- Minimizing employees' training costs.
- Reducing the number of human errors. [8]
- Holding large amount of data/information. [2]
- Centralizing decision making process.

- Making things more efficient by reducing time to solve problems.
- Providing answers for decisions and repetitive tasks. [1]

B. Objectives

- **Reliability and Efficiency:**

  All upgrades and new features that are implemented on the server are delivered automatically to the web-clients. However, maintaining the entire secured database on the server is more reliable and efficient in comparison to customer data on the spreadsheet or physically in the record books. [11]

- **Low on Maintenance:**

  Web browser application using less disk space on the client's system, thereby, amounting to low disk maintenance cost. [16]

- **Platform Independence:**

  Web applications can be integrated into other server-side web procedures easily. [9] It provides cross-platform compatibility (can work on any machine like Mac, Windows, Linux irrespective of the platform used) while they operate within a web browser window. [22]

- **User Friendliness:**

  Most importantly, the application design will have user friendly approach.

- **Easy Accessibility:**

  Records, data and other information can be easily accessed, retrieved or stored. [6]

IV. EXISITING SYSTEM

A. Drawbacks

We are aware of the fact that retail banks are reducing in number [21] due to the demand for online banking.

- **In Multi-National Banks**

  Some banks' websites like HDFC, Citi, Axis and ICICI provide a facility of dynamic FAQ linked to radio buttons and drop-down lists page having different forms linked to each other, but, for getting accurate, precise and desired information, the client finds this facility time consuming. [3]

- **In Smaller Banks**

  It adds up to 3 times employees' salary cost 24x7, 365 days operation plus the training cost which the smaller banks cannot afford, hence, user-friendliness cannot be achieved. [7]

B. Solution

However, incorporating our AI application on bank's website, gives desired and precise information to the web-clients who can ask questions on one-to-one basis.

- **Advantages:**
  1. Has one-to-one interaction between the agent and the web-client.
  2. There is user friendliness.
  3. Is dynamic. [15]
  4. Provides precise results.
  5. Saves time and money.

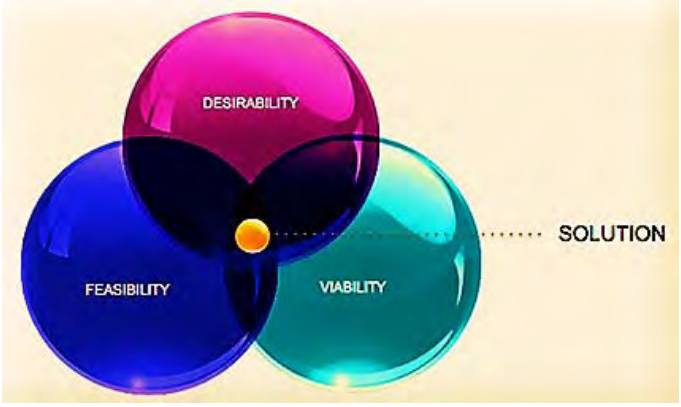

Figure 1. Solution

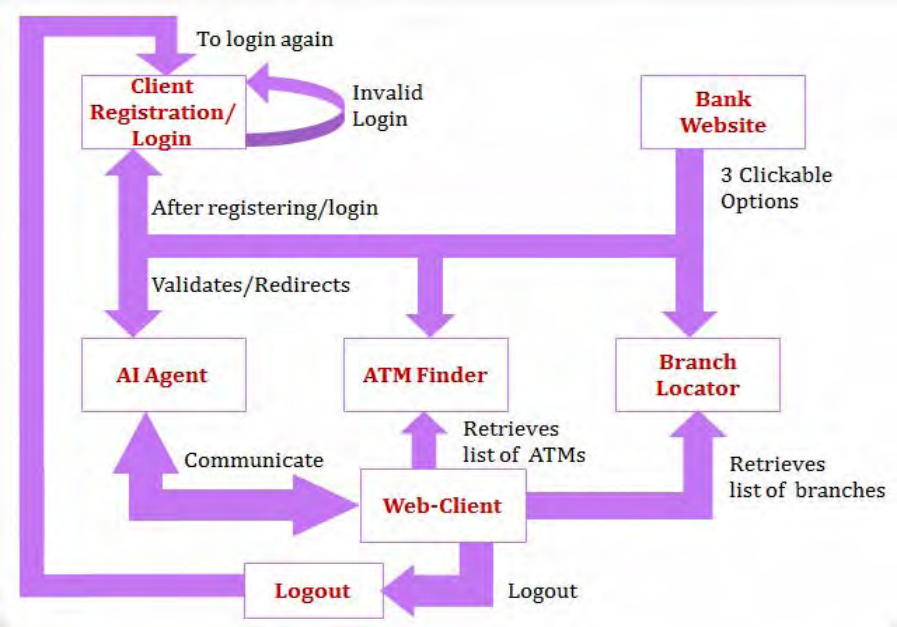

Figure 2. Overview Of Implementation

V. IMPLEMENTATION

A. *Reference Algorithm* [12]

STRING-MATCHING(DB, mins_up):
F1 = {frequent 1-subtree};
F2 = {classes $[P]_1$ of frequent 2-subtrees};
for all $[P1]_1 \in E$ do enumerate-Frequent-Sub_i
ENUMERATE-FREQUENT-SUBTREES([P]):
for each element (x, i) $\in$ [P] **do**
$[P^i_x] = \acute{O}$;
for each element (y, j) $\in$ [P] **do**
Q = {(x, i) x (y, j)};
L(**Q**) = {L(x) x L(y)};
if for any $Q \in$ **Q**, *Q* is frequent
then $[P^i_x] = [P^i_x] \cup \{Q\}$;
Enumerate-Frequent-Subtrees($[P^i_x]$);

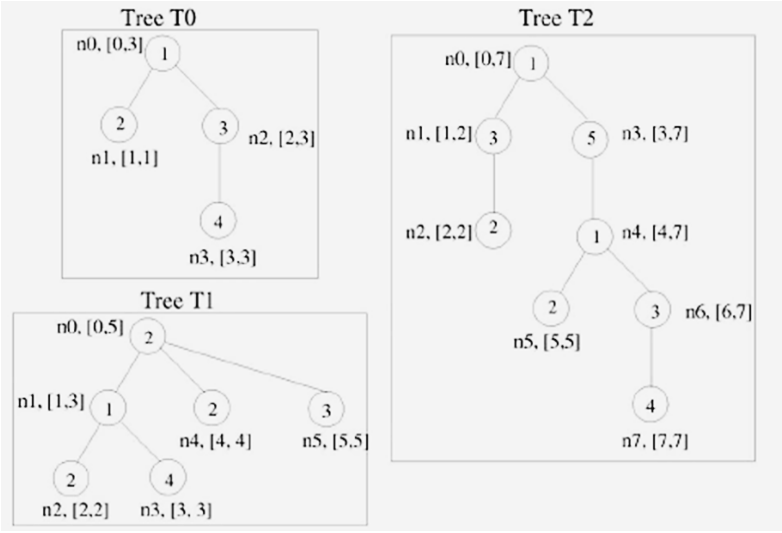

Figure 3. Database 'DB' of Trees - T_0, T_1, T_2

B. Sample Flowchart

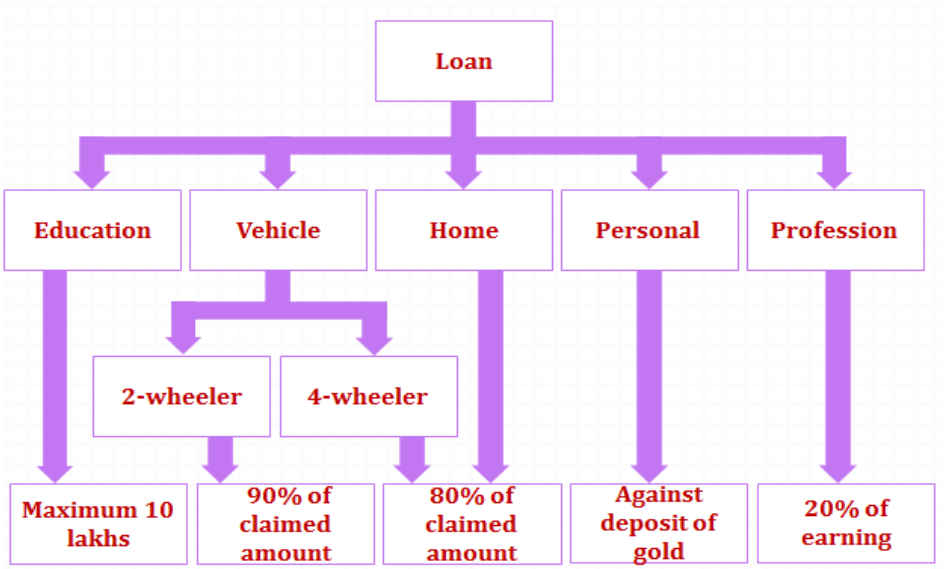

Figure 4. Sample flowchart

C. Implemented System

User friendly AI BANKING AGENT is the interactive dynamic solution with client and is responsive and precise. The client gets desired and precise information in no time, as against the existing system, and banks can use and access our application from the server. Its beauty is simultaneous access by many clients. [16]

1) Modules

a) Effective GUI: A demonstration of how our application would look like, once integrated with Bank's Website has also been implemented in Flash along with the animated talking character that would read out the answers to the queries asked by the web-clients before displaying them on the screen.

b) ATM Finder and Branch Locator: The application also has ATM locator incorporated to help clients find the nearest branch or ATM and find its location. Clients can locate ATM or branch of his interest by searching station or city name; and areas by matching area name.

c) Client Login/Registration: For genuine clients, there is a client login/registration page. If logged in, the client can proceed further else he needs to register, and upon registration, he can login to access the Bank Website and ask queries to the AI agent.

d) Agent: AI will talk as a real banking help desk and provides all the necessary information requested by the web-client.

Here is the Sample Question-Answer format:

Q) *Can you give me information on loan?*
A: *May I know on what loan you need information?*
(Vehicle, Education, Home, Profession).
Q) *I want to know information on vehicle loan.*
A: *May I know on what vehicle loan you need information?*
(Car, Two-wheeler).
Q) *Tell me about car loan.*
A: *Sure. We provide loan on car and it is 30*salary pm.*
Q) *What is the home loan in HDFC Bank?*
A: *Customer Care Number - 18002095363*
Q) *What is Bill Gates doing?*
A: *Sorry, I am not aware of this.*

e) Speech-to-Text: Once the query is asked by the web-client, the animated talking character would read out the answers, and this feature provides the text of the answers spoken out, just in case, the client wants to save it in some document for future use.

2) Implementation Details

a) Software Requirements:

i) *Operating System:* Windows XP/7(professional, ultimate, enterprise)

ii) *SQL Server:* Version 2005/2008 (396 MB)

iii) *Visual studio:* Version 2008/2010/2012 (2.28 GB)

iv) *Macromedia Flash:* Version 2005/2008 (188 MB)

v) *Software Technologies :* Flash, DOT-NET framework

vi) *Programming Languages:* HTML, C#

b) Hardware Requirements:

i) *Processor:* core-2-duo/i3/i5/i7

ii) *Hard Disk Space*: 3 GB (minimum)

iii) *Memory*: 4 GB RAM (in-built)

3) Testing

Figure 5. Effective GUI

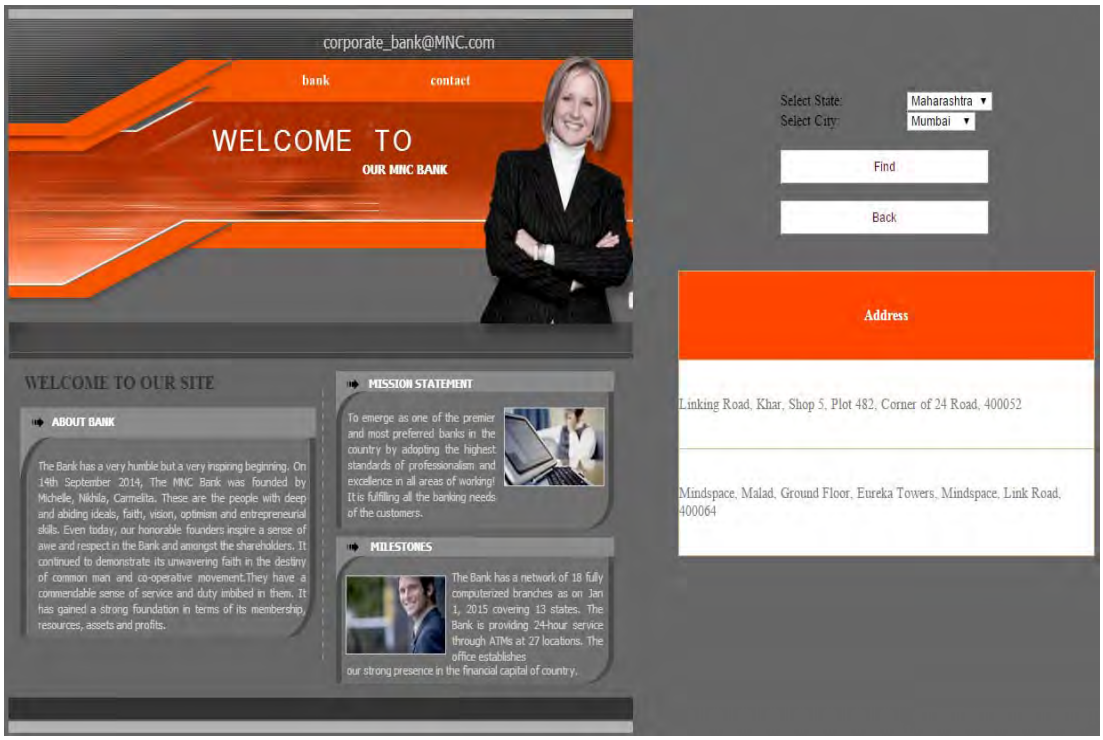

Figure 6. Branch Locator

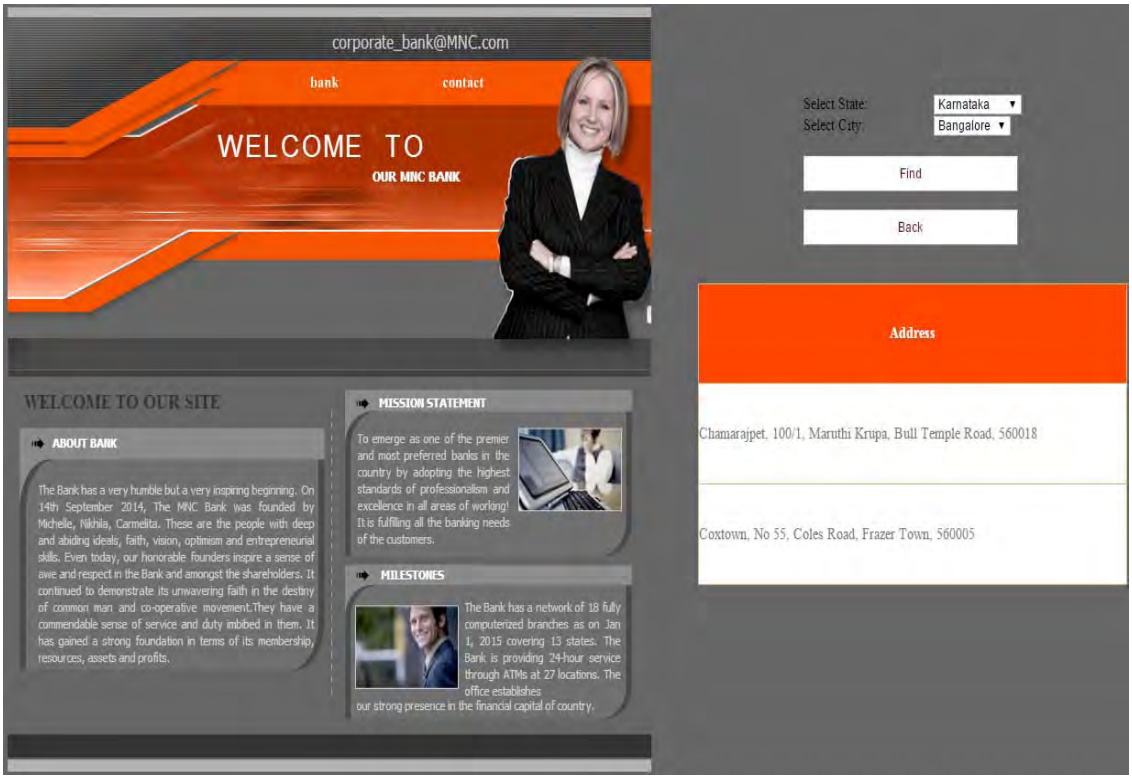

Figure 7. ATM Finder

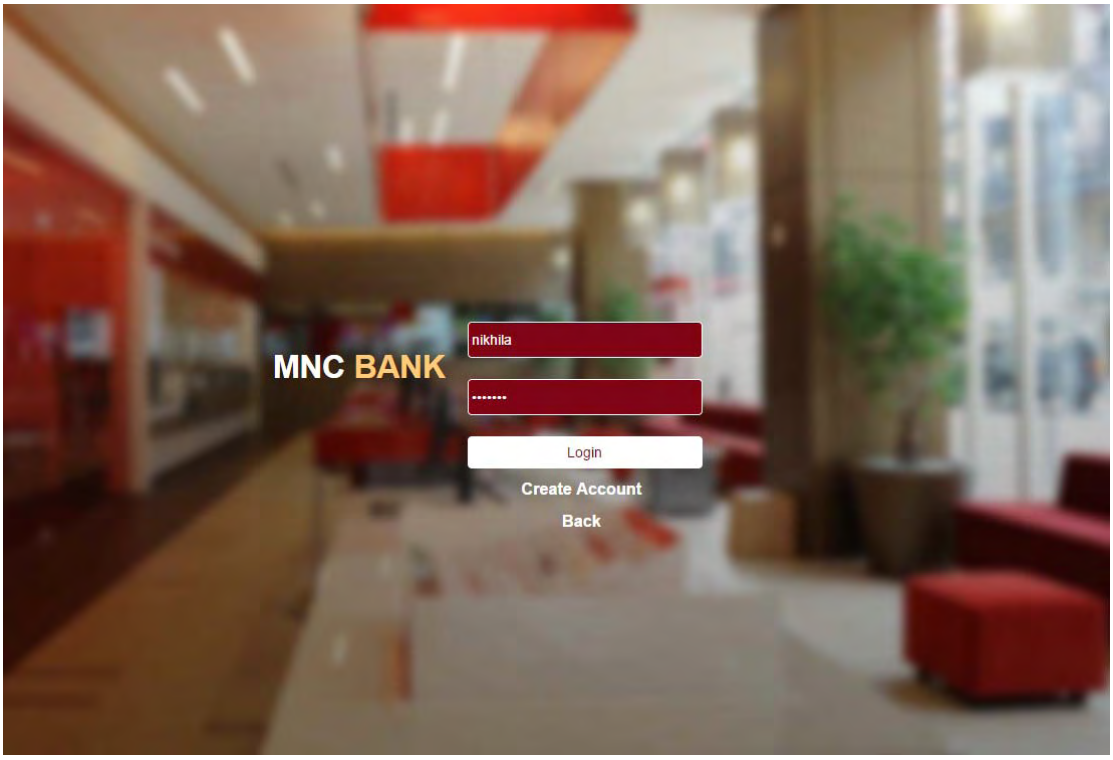

Figure 8. Client Login

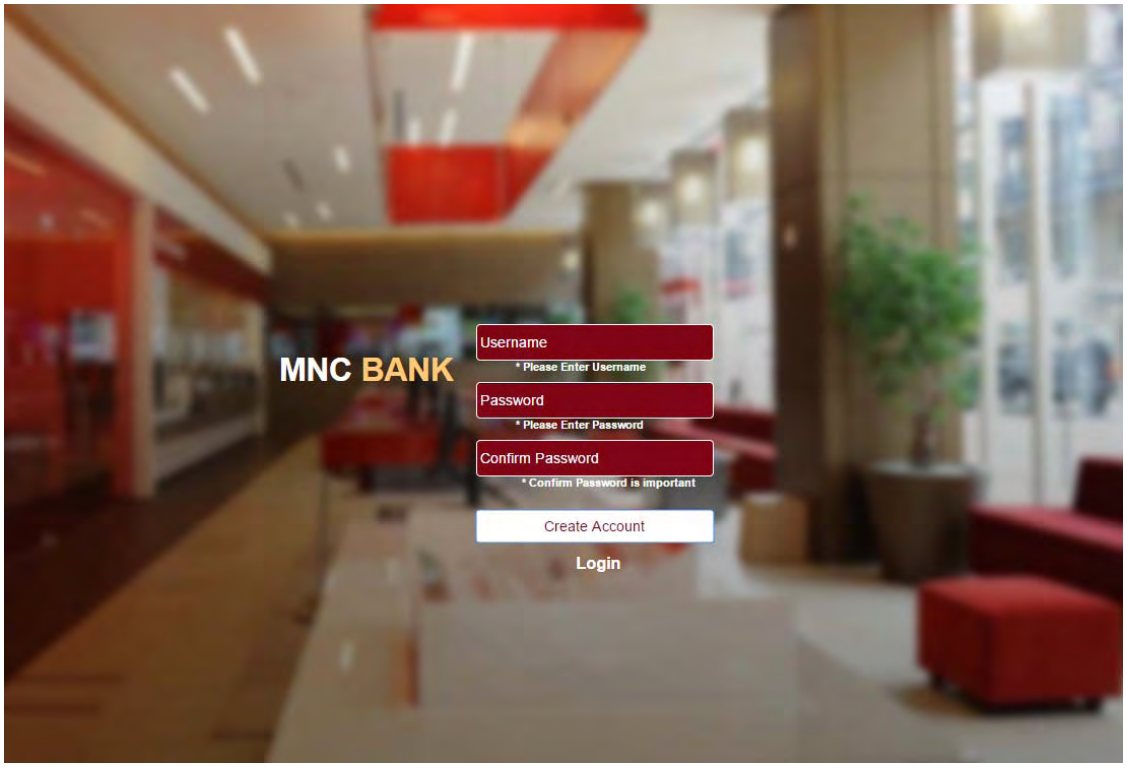

Figure 9. Client Registration

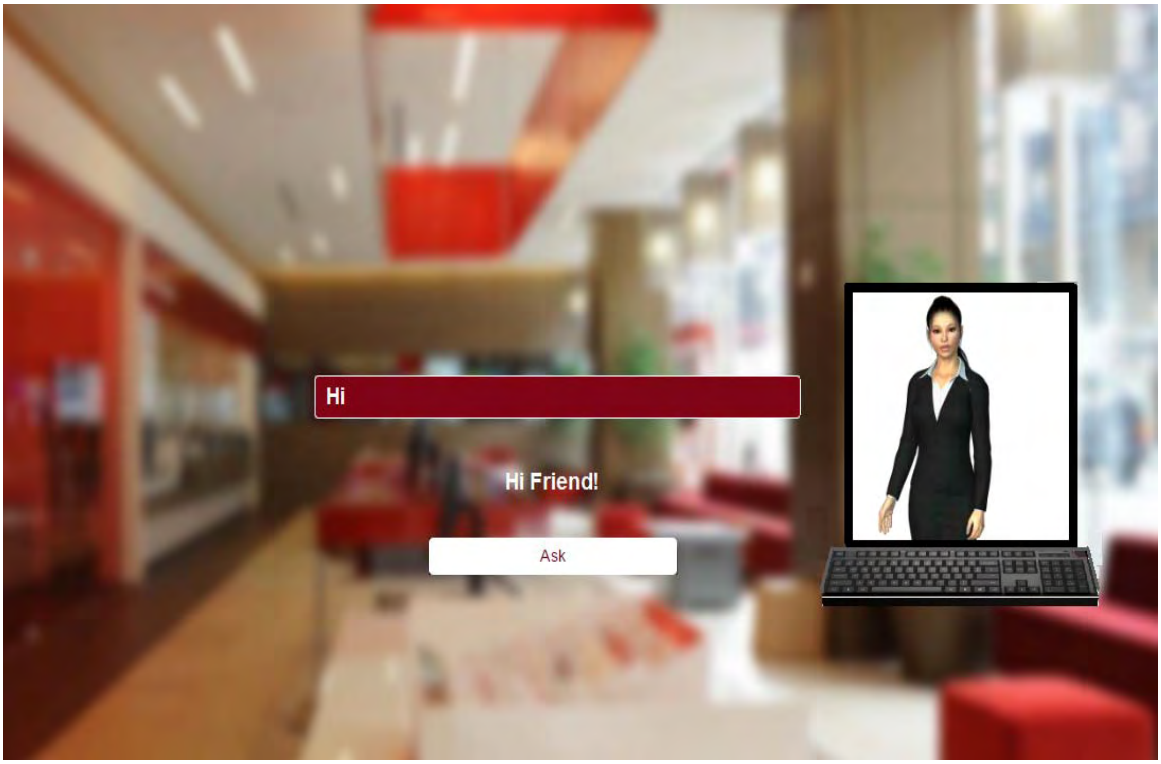

Figure 10. Communication between AI Agent and the Web-Client

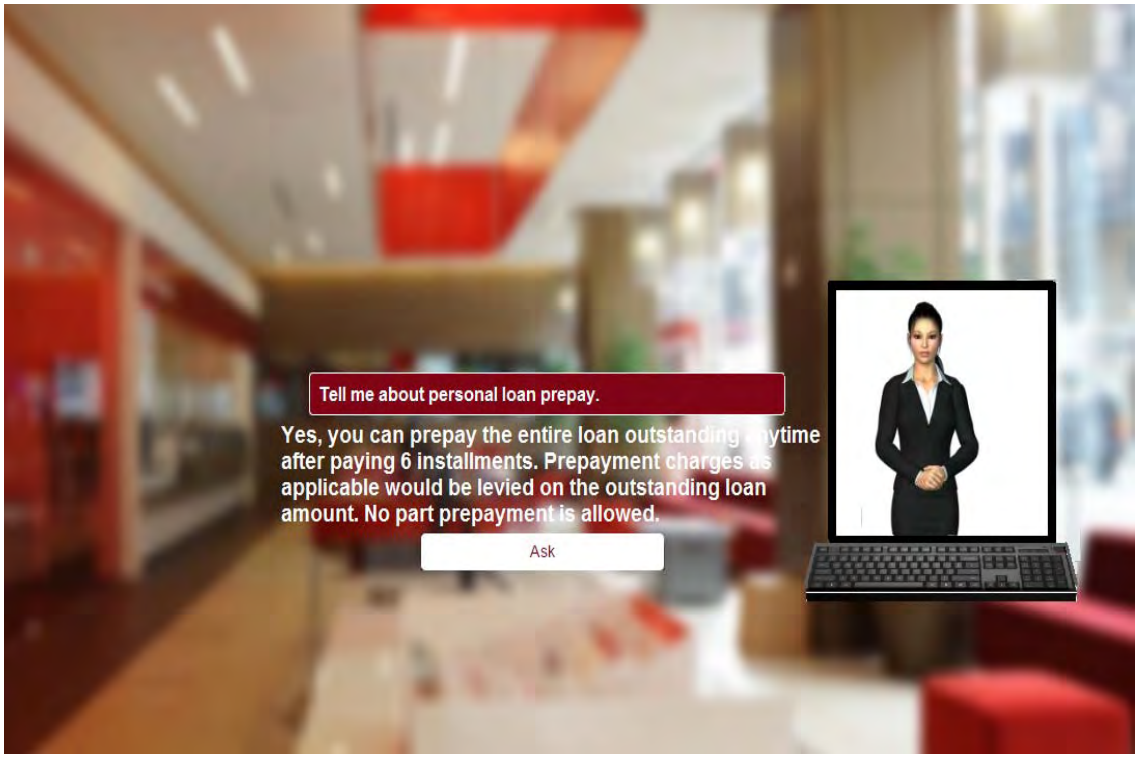

Figure 11. AI Agent providing answers to the queries asked by the Web-Client

VI. CONCLUSION

Artificial intelligence is used in every field and, indeed, very much part of our everyday life as it is helping people to efficiently use information and work smarter. [19] New intelligent applications of AI, which include Agents, are providing new areas of research. In addition to the major role played by them, AI has overcome the difficulties of handling daily routine repetitive services. [13] AI-driven user friendly customer care solutions improved the bottom-line, made customers happy, and freed up the organization resources for

higher-level objectives. [14] However, AI systems are not designed to create an art, but could be, if the need arises, while it is looking to new game genres and even new game paradigms.

REFERENCES

[1] G. Varghese. Fast address look-ups using prefix controlled expansion, Transactions on Computerized Systems, Feb. 1999.
[2] S. Savage, Cristian Estan and G. V. Sumeet Singh, Automated worm fingerprinting. In Proceedings on OS Design and Implementation, Dec. 2003-04.
[3] A. Schwerin, M. Oskin , K. Michelson and S. Swanson. Wavescalar – International Symposium on Microarchitecture, Dec. 2003-02.
[4] J. W. Stevenson and H. Staveren, Using optimization on the intermediate code, Transactions on Programming Language System, 4(1):21–36, 1982.
[5] N. Shnidman, D. Wentzlaff, M. B. Taylora, W. Lee, J. Miller and I. Bratt, Exposed-wire-delay architecture for Streams - Evaluation of the Raw microprocessor, Annual International Symposium on Computerized Architecture, June 2006-07.
[6] G. Varghese, T. Sherwood, B. Calder, N. Tuck - Deterministic memory-efficient string matching algorithms for intrusion detection, IEEE Communications, March 2005.
[7] R. K. Iyer and S. Patel. Architectural support for defending against attack due to buffer overflow, Workshop on To Evaluate and Architect Systems for Dependability, June 2003.
[8] Margaret J. Corasick and Alfred V. Aho, Efficient string matching: an aid to bibliographic search, Communications, v.18 n.6, p.340, June 2003.
[9] G. Relativ Frequency of English Speech Sounds, Dewey Harvard U. Press, p. 185, 1923, Cambridge.
[10] Knuth, D.E., Morris, J.H., and Pratt, V.R. Faster pattern string matching.
[11] Tatiana Tambouratzis, Real-Time Filtering of Large Databases, Proceedings of the International Conference on Adaptive and Natural Computing Algorithms, April 14, 2007, Poland.
[12] Jian Zhang, Theories of Programming and Formal Methods, Springer-Verlag, Berlin, Heidelberg, 2013.
[13] Vincent Alia and S.Sitharama Iyengar, A string searching algorithm, Applied Mathematics and Computation, v.6 n.2, p.123-131, March, 1980-81.
[14] Shaimaa A. El-said, Analysis of secure link state routing protocol, International Journal of Virtual Organisations, v.11 n.1, p.30-47, July 2013.
[15] Miri Kopel Ben-Nissan and Shmuel T. Klein, Boyer Moore Searches on binary texts, Proceedings of the International Conference, July 17-18, 2007, Czech Republic.
[16] Chao-Ching Wang, Chieh-Jen Cheng and Wei-Chun Ku, A high-performance virus detection processor against a large pattern set for embedded network security, IEEE Transactions, v.20 n.5, p.841-854, May 2013.
[17] Rita Hartel and Stefan Bottcher, Efficient string-based XML stream, Proceedings of the Database Conference, p.152, Jan 31-Feb 03, 2012, Australia.
[18] Nai-Lun Huang and Tsern-Huei Lee, A string-matching scheme with low memory requirement and high throughput performance, IEEE Transactions, v.21, p.1104-1116, August 2013.
[19] Jorma Tarhio and Kjell Lemstrom, Transposition pattern matching for multi-strings, Nordic Journal of Computing, v.10 n.3, p.185-205, Sept 2003.
[20] Christof Paar and Daehyun Strobel, An efficient method to eliminate delays in power traces of embedded system software, In the Proceedings of the international conference on Information Security, December 02, 2013, Korea.
[21] Paulo Carvalho and Sergio Deusdado, The Exact Efficient Pattern-Matching, Proceedings of the 10th International Conference on Soft Computing - Artificial Neural Networks, Ambient Assisted Living, Distributed Computing, Bioinformatics and Artificial Intelligence, June 12, 2009, Spain.
[22] Kuramoch and Karypis, Members of IEEE, An efficient Algorithm for Discovering Frequent Sub-Graphs, IEEE Transaction Knowledge and Data Engineering, September 2004.