Home Rental System Implementing Constraint Satisfaction Problem

By

Nusrat Momtahana (13101075)

Nawshin Tabassum (13101070)

Thesis directed by

Dr Amitabha Chakrabarty

Department of Computer Science and Engineering

BRAC University

This dissertation is submitted for the degree of

Bachelor of Science in Computer Science

We would like to dedicate this thesis to our parents...

Declaration

We hereby declare that this report is our original work and that has not been submitted anywhere for any award. Materials of work found by other researchers are mentioned with due reference. All the contents provided here is totally based on our own effort dedicated for the completion of the thesis.

The work is done under the guidance of Dr. Amitabha Chakrabarty, at the Department of Computer Science and Engineering, BRAC University, Dhaka.

Date of Submission: 14th December, 2016

Signature of the supervisor

_____

Dr. Amitabha Chakrabarty

Assistant professor

Department of Computer Science and

BRAC University

Signature of Author

_____

Nawshin Tabassum

_____

Nusrat Momtahana

Abstract

An efficient and dynamic home rental system which users can access through their mobile phone with no hassle is a desire of every tenant in Bangladesh. In this era of technology no one want to roam around to find a vacant house. House for living is one of the basic needs of people, therefore to make it easier and accessible by every people we represent a home rental system which provides every needed facility .CSP(Constraint Satisfaction Problem) implemented search option is more dynamic in case of large database compared to SQL (Structured Query Language) search. The primary focus of our work is to implement constraint satisfaction problem in the search option of our home rental system. Our home rental system will have dynamic values for searching and for which constraint satisfaction problem implementation is a better preference. Not only this but also a combination of public transport tracker and a dynamic chat server between admin and agent is also a feature of this rental system. Additionally, there is a dynamic mail alert system in which if any user put any request in the wish list, they will be notified by this system. We implement IOT (Internet of Things) in one small option. That is, to detach family and bachelor criteria. Here we discuss how we program our search feature using CSP search based algorithm and also the implementations of transport tracker, dynamic mail alert and the chat server.

Acknowledgements

We would like to show our gratitude to almighty Allah at first. We got so many friends' support through the completion of our thesis work, such words will not be enough to thank them.

We would like to express our gratification to our Supervisor Dr. Amitabha Chakrabarty for being constant source of inspiration, valuable guidance and constant encouragement to us especially for solving the problems that we have encountered while working on this thesis.

We are also grateful to Ahmad Naquib Chowdhury, Tahmid Tanzi Alam, Nura Jamil- students of BRAC University for being with us throughout the thesis work.

Last but not least, we would like to thank our parents and family members for all their trust and support.

Table of Contents

List of Figures

List of Algorithm

List of Tables

Chapter 1

Introduction

According to Rossi, van Beek and Walsh, CSP constitute an important formalism of Artificial Intelligence (AI) for expressing and efficiently solving a wide range of practical problems [2]. Therefore, the development of effective solution techniques for CSP's is an important research problem in the field of dynamic and complex system. Hence CSP is dynamic; it gives more accurate result in home rental system. We worked with CSP in such home rental system to build a more dynamic system than the existing ones. Our goal of this thesis is to make the system as user-friendly as possible. For this purpose, we have shrink the whole in via an icon in the mobile phones as well so that the users can search for vacant houses and transport at any time. To minimize any kind of communication gap, we introduce a chatting system between agents and admin. Users always want to get a result in the shortest span of time. In many cases, may be users get the reply lately due to internal long manual processes. To solve that, we introduce a dynamic mail system to notify the users' search result in the least possible time. Therefore, the home rental system hold features like rental search, dynamic mail system, tenants can search for vacant houses randomly according to their wishes based on unique area or city, they can track for vehicles from their given location as well. Additionally, they can add their wish list if their desired home is not available. There will be a chat system in which admin and agent can chat directly. All the values which are used for processing result are dynamic in our home rental system. CSP can give the result by creating domain based on the constraints. In our system, there is an option whether u want family house or bachelor, we implement IOT here to make more efficient. In Dhaka city, everyday so many new residents enter for job or study. However, recently it has become a trivial issue for the bachelors because none of the house owners are

ready to rent their houses to them. Therefore, we create such a safe and user friendly platform for the bachelors so that they can rent house efficiently and in a hassle free procedure.

1.1 Problem Definitions:

Roam around to rent a house has always been a hassle for people. Especially, on recent times, people have so many priorities based on which they have to rent their house. Some people want their house to be in the commercial space, or some want in a chaos free space. Some people prefer to choose the area of their house relating the religion they belong. Again there are a lot of people who love pets; therefore they want a house which has pet allowance. Basically, in this era of modernism people want to rent their house like online shopping. To rent a house in physical world has become less popular now a days [1]. No one wants to roam around here and there to search for a house. People would prefer a virtual system to rent a house.

To decode this situation and to represent a hassle free environment to the people, a dynamic system can be implemented. That system would give the tenants the best service for renting houses without any kind of hassle. Government can make one unique system where people can rent house based on their priority instead of having so many rental systems. In that system, all the vacant houses of any district of Bangladesh will be listed there. One system will hold every details of every vacant house from any district, any are. To, make the system more liable, there should be a system by which tenants can verify the owner or agent. Also to analysis the place they will rent for house they need to know the location of that. Hence, every information details which have minimum priority to rent a house will hold by the system. There a one special feature for the bachelors so that they can rent houses efficiently as now a day house owners do not want to rent their houses to the bachelors for safety issue.

1.2 Motivation:

According to the universal declaration of human rights by which we get to know that is belong to a human the right to have a proper standard of living. (United Nations, The Human Rights-article 25, 1948).[1] A standard living place is every citizen's basic right. .Choosing that place is also their right. As technology is growing so fast every single day, it is necessary to make the system the most dynamic approach. By the need of this, a dynamic home rental system is needed where every citizen can choose their house according to their choice and also it has to be most dynamic as people will access this system at anytime from anywhere.

To make the system more dynamic we have implemented CSP algorithm in the following way. CSP make the search more dynamic then SQL. In CSP, first a priority can be set. Then user will give constraints, based upon the constraints and a priority domain will be created using the available data. The domain is mainly all the rentals related to the constraint set on the priority filed. Hence, users can give any constraints and also set any priority; CSP will create the initial priority based domain and from that domain it will lead towards the result domain after handling other sub domains which will be created from the priority domain relating the fields which are not set as priority. Suppose in our system user can set the priority by area and rent range, then he/she will give their constraints

As a result, it is such a system which will represent a dynamic approach of renting home.

1.3 Chapter Layout

The paper is divided into 5 sections:

- Section 2 gives an overview of the literature review.
- Section 3 includes implementation of csp algorithm in our home rental system.
- Section 4 describes the possible experimental result.
- In section 5, we conclude our thesis work and mention the future work we are heading towards.
- Lastly and chapter 6 we cited the references with proper citation method.

Chapter 2

Literature Review

In this thesis, we studied numerous algorithm to make sure the applied one will be the dynamic and efficient one. We researched on constraint satisfaction problem algorithm, sql search algorithm, brute force algorithm, self organizing algorithm, binary search algorithm, incremental search algorithm. After studying all these we came up to a point that we should work with csp. As csp can work with huge database and also can manage data dynamically. We implemented csp in the search option but our system provide other features also. Therefore to keep pace with every feature we have to implement that algorithm with works dynamically and that is why we chose csp. We represent brief explanation of all the algorithms we researched for our thesis the following paragraphes.

2.1 Constraint Satisfaction Problem (CSP):

A problem of CSP can be derived in the following process. Firstly, there will be a given set of variables and also a finite set of possible values which can be assigned to each variable. Additionally, a list of constraints will also be there. CSP will find values of the variables which satisfy every constraints from the list. [4]

According to Liu, Forward Checking (FC) with some other heuristics has been traditionally considered to be the best algorithm for solving CSPs while in recent time there have been a number of claims that maintaining arc consistency (MAC) is more efficient on large and hard CSPs. [5]

Hence, many experiment for example configuration and model composition, the set of variables which are pertinent to a solution and that has to be assigned value changes dynamically in retort to decisions taken during the course of problem response.[6]

The concept of CSP algorithm can be summarized by,

A constraint satisfaction problem is formally defined as:

- A set of variables which can be derived as $x_i \ldots x_n$
- Each of the variable has a domain of values it can take as $Di_i \ldots D_n$
- A set of constraints $C_i \ldots C_n$ which specifies acceptable combinations of values for a subset of the variables.[4]

  To make it clearer we can define CSP using some equation and flow chart in the following paragraph.

  First, it takes a set of variables,

  $WV = \{(w_1 v_1),(w_2 v_2), \ldots, (w_n v_n)\}$, c are entered (1)

  Then there will a bin which is mainly a set of problem lists, is set to 0,

  $BS=0$ (2)

  BS is the number of bins or set of problems .Then a problem have to solve by CSP first it will check for the range of variables :

  $\int_{f=1}^{n} w_f \, v_f > c$ (3)

  If this is yes then, it will solve for a bin. The solution is denoted by :

  $(k_1, k_2, \ldots, k_n)$ (4)

  k_i shows how many item a_i is placed into the bin.

  Then a maximum range for the set of solution of bins will be determined,

max $\{\frac{k_i}{v_i}\}=\frac{k_p}{v_p}$ (5)

Taking $v_p = 1 - k_p$ is resolved for the bin. Large sized items used more are acceptable,

$(k_1', k_2', \ldots, k_n')$ (6)

Then a solution from the set of bins will be found and these will change,

$min\{\frac{v_i}{k_i}\}=t$ (7)

$v_{i=}v_i\text{-}k_i't$ (8)

$BS=BS+t$ (9)

The the set of bins will be incremented,

$BS=BS+1$ (10)

This is how result will be found out .

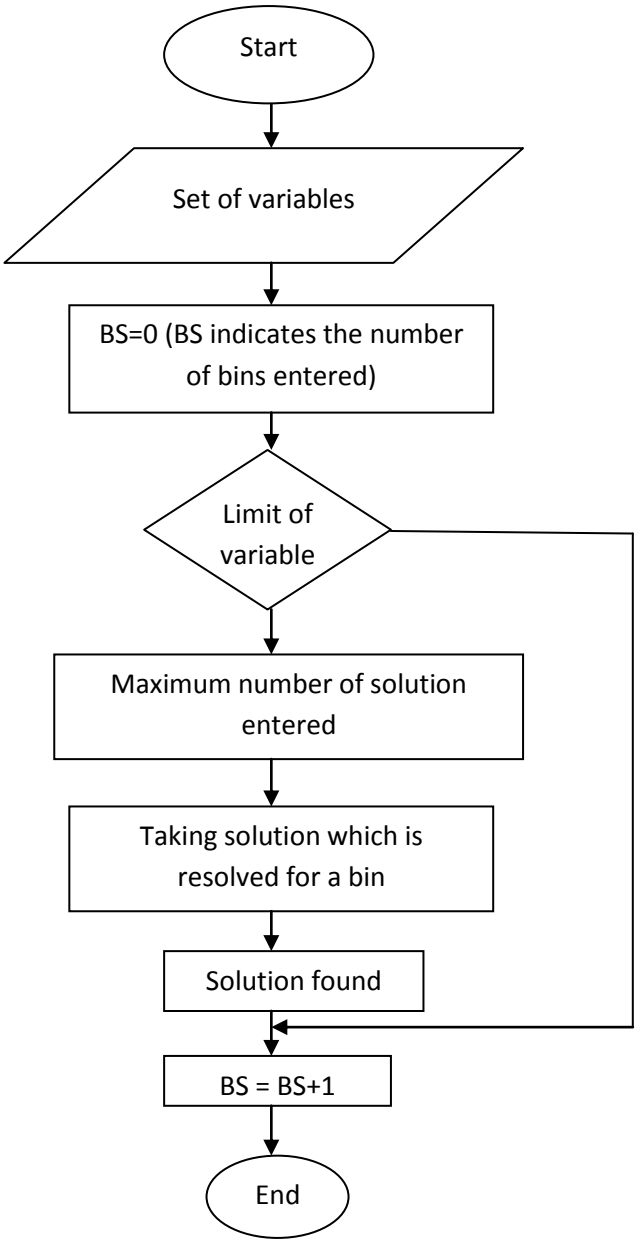

Figure 2 1: Flow Chart of CSP Architecture

2.1.1 CSP Applications:

Problems like scheduling such as time tabling problem uses CSP in solution. The problem of scheduling can be expressed by assigning a set of experiments to a set of resources subject to a set of constraints.[7]

Apart from scheduling blocks of time there are some other real world applications in which CSP has been implemented such as Sudoku, Coloring maps.[8]

The entire concept can be summarized in the following figure:

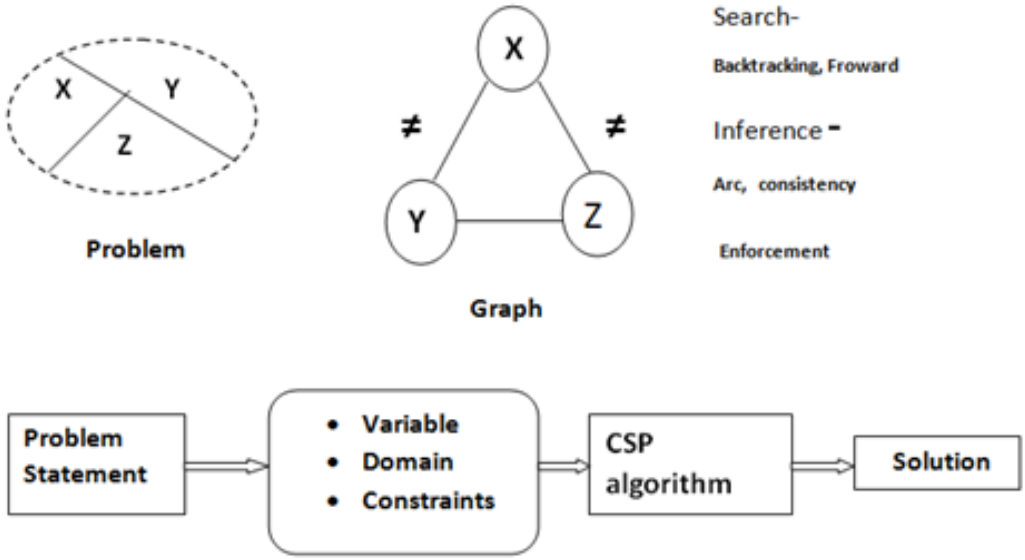

Figure 2 2: CSP Representation

2.1.2 Backtracking Algorithm in CSP :

Definition:

This section introduces the notational framework used throughout the paper. Propositional variables are denoted xl,..., xn, and can be assigned truth values 0 (or F) or 1 (or T). The truth value assigned to a variable x is denoted by v(x). A literal I is either a variable x~ or its negation ┐xi. A clause ω is a disjunction of literals and a CNF formula ζ is a conjunction of clauses[18]. A clause is said to be satisfied if at least one of its literals assumes value 1, unsatisfied if all of its literals assume value 0, unit if all but one literal assume value 0, and unresolved otherwise. [19]. Literals with no assigned truth value are said to be free literals. A formula is said to be satisfied if all its clauses are satisfied, and is unsatisfied if at least one clause is unsatisfied.[20]

2.1.3 Forward checking:

The backtracking algorithm only checks the constraints between the current variable and the past variables.csp8 On the other hand, forward checking and MAC are look ahead algorithms that check the constraints between the current and past variables and the future variables. [11] In the forward checking algorithm, when a value is assigned to the current variable, any value in the domain of a future variable which conflicts with this assignment is (temporarily) removed from the domain. [21] The advantage of this is that if the domain of a future variable becomes empty, it is known immediately that the current partial solution is inconsistent, and as before, either another value for the current variable is tried or the algorithm backtracks to the previous variable; the state of the domains of future variables, as they were before the assignment which led to failure, is restored[22]. With simple backtracking, this failure would not have been detected until the future variable was considered, and it would then be discovered that none of its values are consistent with the current partial solution. Forward checking, therefore, allows branches of the search tree that will lead to failure to be pruned earlier than with simple backtracking.[23]

2.2 SQL Search:

SQL can be defined as Structured Query Language. SQL is the most commonly used database query language [3]. In 1970, SQL was developed in IBM Research Laboratories. It includes retrieval, manipulation, and administration of data stored in tabular form. SQL follows such principles in designing:

- SQL is nonprocedural and a high level language intended foe processing by an optimizing compiler.
- SQL is intended to be accessible to users without formal training in mathematics or computer
  programming. It is designed to be typed on a keyboard. Therefore it is framed in familiar English
  keywords, and avoids specialized mathematical concepts or symbols.

- SQL attempts to unify data query and update with database administration tasks such as creating and modifying tables and views, controlling access to data, and defining constraints to protect database integrity. In pre-relational database systems, these tasks were usually performed by specialized database administrators and required shutting down and reconfiguring the database.[24] By building administrative functions into the query language, SQL helps to eliminate the database administrator as a choke point in application development.
- SQL is designed for use in both decision support and online transaction processing environments. The former environment requires processing of complex queries, usually executed infrequently but accessing large amounts of data.[3]

2.2.1 SQL Queries:

SQL operation occurs in form of tables using data. Every individual table has a name and consists of one or more columns, each of the table has a name and a data type. The table consists of zero or more rows. The value associated with a given row and column may be an

instance of the datatype of that column, or may be a special "null" value indicating that the value is missing (not available or not applicable). SQL statements, which may be queries or updates, operate on stored tables or on tabular "views" that are derived from stored tables. The result of a query is an unnamed virtual table. The result of an update is a change to the stored data, which is visible to subsequent statements. Generally, updates can be applied to a view only if each row in the view can be mapped uniquely onto a row of a stored table. The following rule indicates to map every update on the view to updates on the underlying table. [3]

SQL includes some key points for example RDBMS, Table, Field, Record or Row, Column, Null Value,

- RDBMS:
  RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access. A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

- Table:

- The data in RDBMS is stored in database objects called tables. The table is a collection of related data entries and it consists of columns and rows.

- Remember, a table is the most common and simplest form of data storage in a relational database. Following is the example of a CUSTOMERS table:

ID	NAME	AGE	ADDRESS	SALARY
1	Nawshin	25	Uttara	5000.00
2	Nusrat	30	Niketon	6000.00
3	Rafi	26	Baridhara	7000.00

Table 1: Example of table in SQL query

- **Field:**

Every table is broken up into smaller entities called fields. The fields in the CUSTOMERS table consist of ID, NAME, AGE, ADDRESS and SALARY. A field is a column in a table that is designed to maintain specific information about every record in the table.

- **Record or Row:**

A record, also called a row of data, is each individual entry that exists in a table. For example there are 7 records in the above CUSTOMERS table. Following is a single row of data or record in the CUSTOMERS table:

1	Nawshin	25	Uttara	5000.00

Table 2: Example of row in SQL query

A record can be specified as a horizontal entry in the table.

- **Column:**

A column is a vertical entity in a table that contains all information associated with a specific field in a table. For example, a column in the CUSTOMERS table is ADDRESS, which represents location description and would consist of the following:

ADDRESS
Uttara
Niketon
Baridhara

Table 3: Example of column in SQL query

- **Null Value:**

A NULL value in a table is a value in a field that appears to be blank, which means a field with a NULL value is a field with no value. It is very important to understand that a NULL value is different than a zero value or a field that contains spaces. A field with a NULL value is one that has been left blank during record creation.[14]

2.2.2 Constraints:

Constraints are the rules enforced on data columns on table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints could be column level or table level. Column level constraints are applied only to one column where as table level constraints are applied to the whole table.

Following are commonly used constraints available in SQL:

- NOT NULL Constraint: Ensures that a column cannot have NULL value.
- DEFAULT Constraint: Provides a default value for a column when none is specified.
- UNIQUE Constraint: Ensures that all values in a column are different.

- PRIMARY Key: Uniquely identified each rows/records in a database table.
- FOREIGN Key: Uniquely identified a rows/records in any another database table.
- CHECK Constraint: The CHECK constraint ensures that all values in a column satisfy certain conditions.
- INDEX: Use to create and retrieve data from the database very quickly.

2.2.3 Data Integrity:

There are some types of integrities with each RDBMS:

- Entity Integrity: There are no duplicate rows in a table.
- Domain Integrity: Enforces valid entries for a given column by restricting the type, the format, or the range of values.
- Referential integrity: Rows cannot be deleted, which are used by other records.
- User-Defined Integrity: Enforces some specific business rules that do not fall into entity, domain or referential integrity.[14]

2.2.4 Database Normalization:

This is such a process where data is organized efficiently in a database. Two reasons can be provided of doing normalization:

- Eliminating redundant data, for example, storing the same data in more than one table.
- To ensure data dependencies is logical.[26]

Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored. Normalization consists of a series of guidelines that help guide you in creating a good database structure.

Normalization principles are divided into normal forms. We can think it as the format or the way a database structure is laid out. The aim of normal forms is to organize the database structure so

that it complies with the rules of first normal form, then second normal form, and finally third normal form.

It's your choice to take it further and go to fourth normal form, fifth normal form, and so on, but generally speaking, third normal form is enough.[25]

- First Normal Form(1NF)
- Second Normal Form(2NF)
- Third Normal Form(3NF)

The last form which is 3NF is the most simplified form and generalized one.

2.2.5 MySQL:

Among a lot of mostly used RDBMS exist to work with; MySQL is another one. It is an open Source SQL database[27]. This is developed by Swedish company MySQL AB. MySQL is pronounced “my ess-que-ell”,in contrast “sequel”. MySQL is supporting many different platforms including Microsoft Windows, the major Linux distributions, UNIX, and Mac OS X. MySQL has free and paid versions, depending on its usage (non-commercial/commercial) and features. MySQL comes with a very fast, multi-threaded, multi-user, and robust SQL database server.[14]

Features:

- High Performance.
- High Availability.
- Scalability and Flexibility Run anything.
- Robust Transactional Support.
- Web and Data Warehouse Strengths.
- Strong Data Protection.
- Comprehensive Application Development.
- Management Ease.
- Open Source Freedom and 24 x 7 Support.

- Lowest Total Cost of Ownership.[14]

2.3 Incremental Search Algorithm:

Incremental search algorithms, such as D* Lite, reuse information from previous searches to speed up the current search and can thus solve sequences of similar search problems faster than Repeated A*, which performs repeated A* searches. [13]

So far, three main classes of incremental heuristic search algorithms have been developed:

- The first class restarts A* at the point where its current search deviates from the previous one (example: Fringe Saving A*).
- The second class updates the h-values from the previous search during the current search to make them more informed (example: Generalized Adaptive A*).
- The third class updates the g-values from the previous search during the current search to correct them when necessary, which can be interpreted as transforming the A* search tree from the previous search into the A* search tree for the current search (examples: Lifelong Planning A*.

All three classes of incremental heuristic search algorithms are different from other re-planning algorithms, such as planning by analogy, in that their plan quality does not deteriorate with the number of re-planning episodes. [12] Incremental heuristic search has been extensively used in robotics, where a larger number of path planning systems are based on either D* (typically earlier systems) or D* Lite (current systems), two different incremental heuristic search algorithms.

2.4 Binary Search:

All of the sequential search algorithms have the same problem; they walk over the entire list. Some of our improvements work to minimize the cost of traversing the whole data set, but those improvements only cover up what is really a problem with the algorithm. By thinking of the data in a different way, we can make speed improvements that are much better than anything sequential.

Consider a list in ascending sorted order. It would work to search from the beginning until an item is found or the end is reached, but it makes more sense to remove as much of the working data set as possible so that the item is found more quickly. If we started at the middle of the list we could determine which half the item is in (because the list is sorted). This effectively divides the working range in half with a single test [11]. By repeating the procedure, the result is a highly efficient search algorithm called binary search. The actual algorithm is surprisingly tricky to implement considering the apparent simplicity of the concept. Binary search is very efficient, but it can be improved by writing a variation that searches more like humans do. Consider how you would search for a name in the phonebook. I know of nobody who would start in the middle if they are searching for a name that begins with B. They would begin at the most likely location and then use that location as a gauge for the next most likely location. Such a search is called interpolation search because it estimates the position of the item being searched for based on the upper and lower bounds of the range. Interpolation search is theoretically superior to binary search. With an average time complexity of O (log log n), interpolation search beats binary search's O(log n) easily. However, tests have shown that interpolation search isn't significantly better in practice unless the data set is very large. Otherwise, binary search is faster. Moreover, the algorithm is not efficient for solving constraint satisfaction problem.

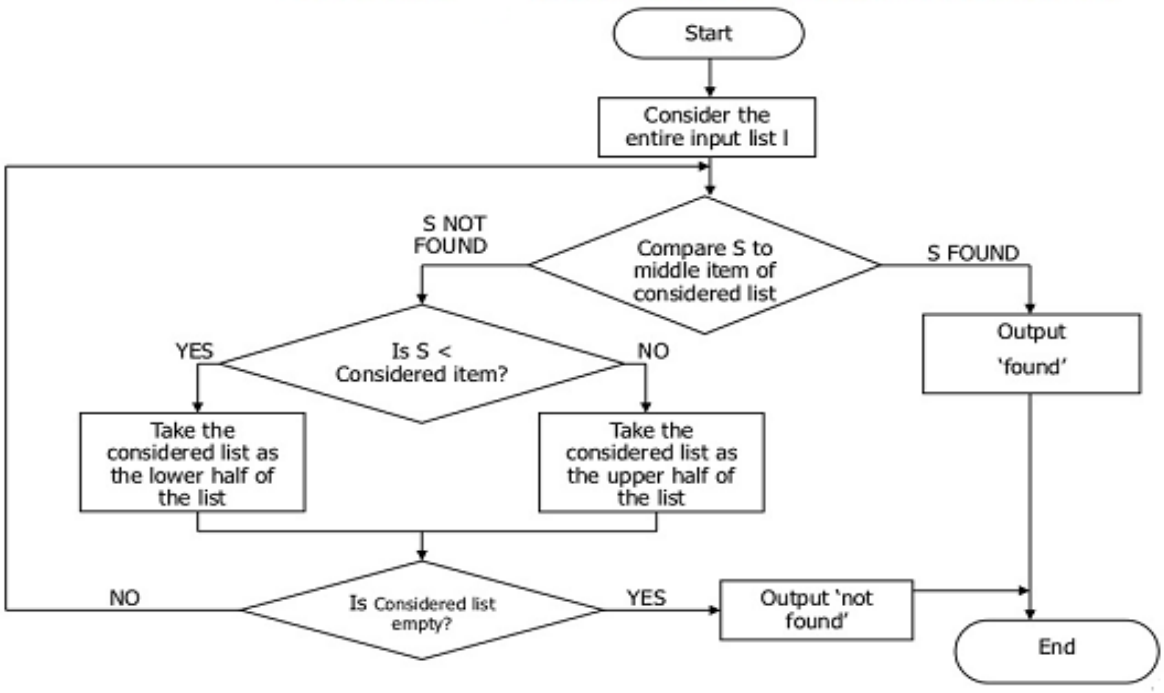

Figure 2 3: Flow chart of Binary Search

2.5 Self Organizing Algorithm:

This algorithm is a type of neural network. For lists that do not have a set order requirement, a self organizing algorithm may be more efficient if some items in the list are searched for more frequently than others[16]. By bubbling a found item toward the front of the list, future searches for that item will be executed more quickly. This speed improvement takes advantage of the fact that 80% of all operations are performed on 20% of the items in a data set. If those items are nearer to the front of the list then search will be sped up considerably. [10] As this not efficient to solve Constraint Satisfaction Problem efficiently, we did not use it.

2.6 Brute Force Algorithm:

In computer science, brute-force search or exhaustive search, also known as generate and test, is a very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.
[9]

A brute-force algorithm to find the divisors of a natural number *n* would enumerate all integers from 1 to n, and check whether each of them divides *n* without remainder. A brute-force approach for the eight queens puzzle would examine all possible arrangements of 8 pieces on the 64-square chessboard, and, for each arrangement, check whether each (queen) piece can attack any other.

While a brute-force search is simple to implement, and will always find a solution if it exists, its cost is proportional to the number of candidate solutions – which in many practical problems tends to grow very quickly as the size of the problem increases. Therefore, brute-force search is typically used when the problem size is limited, or when there are problem-specific heuristics that can be used to reduce the set of candidate solutions to a manageable size[17]. The method is also used when the simplicity of implementation is more important than speed.

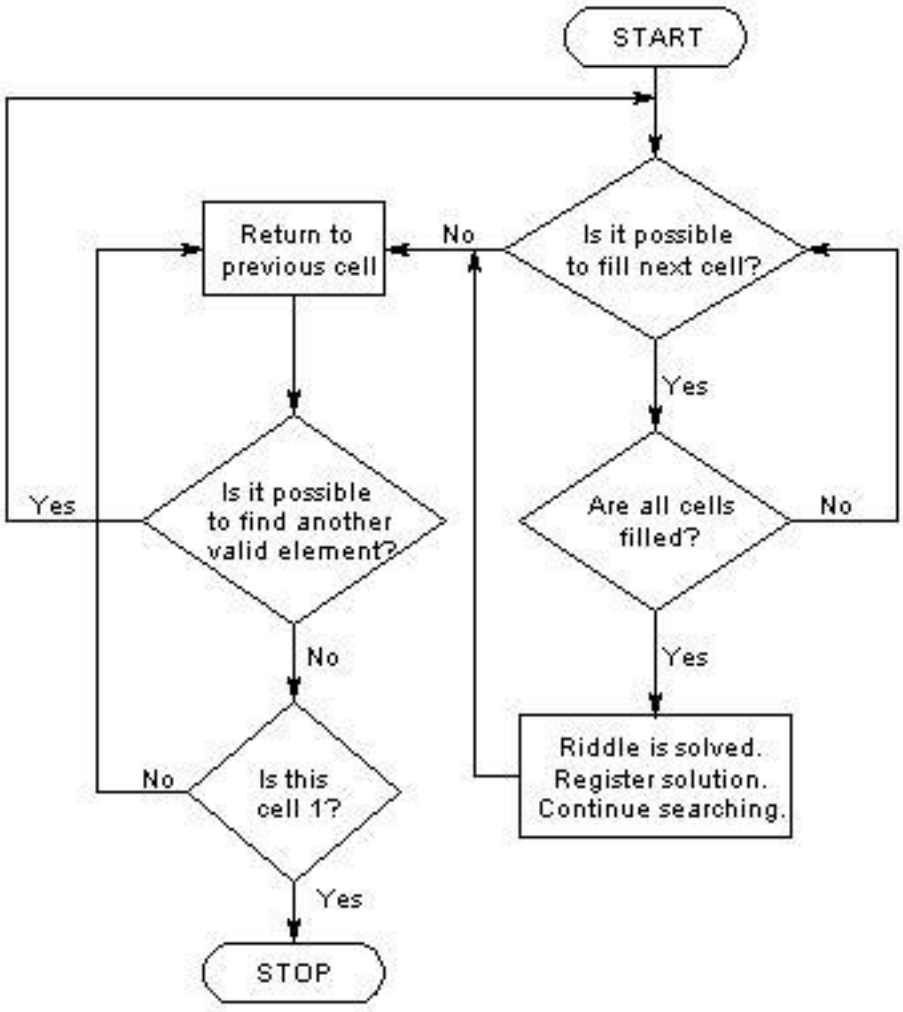

Figure 2 4: Flow chart of Brute Force Algorithm

2.7 Ajax:

AJAX, is a web development technique for creating interactive web applications. If you know JavaScript, HTML, CSS, and XML, then you need to spend just one hour to start with AJAX. AJAX stands for **A**synchronous **Ja**vaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script. Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display. Conventional web applications transmit

information to and from the server using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server. With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server. XML is commonly used as the format for receiving server data, although any format, including plain text, can be used. AJAX is a web browser technology independent of web server software. A user can continue to use the application while the client program requests information from the server in the background. ntuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger. Clicking is not required, mouse movement is a sufficient event trigger. Data-driven is as opposed to page-driven. AJAX is based on the following open standards:

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).
- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen.[15]

AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.

JavaScript

- Loosely typed scripting language.
- JavaScript function is called when an event occurs in a page.
- Glue for the whole AJAX operation.

DOM

- API for accessing and manipulating structured documents.
- Represents the structure of XML and HTML documents.

CSS

- Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript.

XMLHttpRequest

- JavaScript object that performs asynchronous interaction with the server.

Ajax has been used in the following web applications:

- Google Maps
- Google Suggest
- Gmail
- Yahoo Maps

Ajax support this following browsers:

- Mozilla Firefox 1.0 and above.
- Netscape version 7.1 and above.
- Apple Safari 1.2 and above.
- Microsoft Internet Explorer 5 and above.
- Konqueror.
- Opera 7.6 and above.

2.7.1 Steps of AJAX Operation

- A client event occurs.
- An XMLHttpRequest object is created.
- The XMLHttpRequest object is configured.
- The XMLHttpRequest object makes an asynchronous request to the Webserver.
- The Webserver returns the result containing XML document.

- The XMLHttpRequest object calls the callback() function and processes the result.
- The HTML DOM is updated.

The concept can be summarized in the following figure:

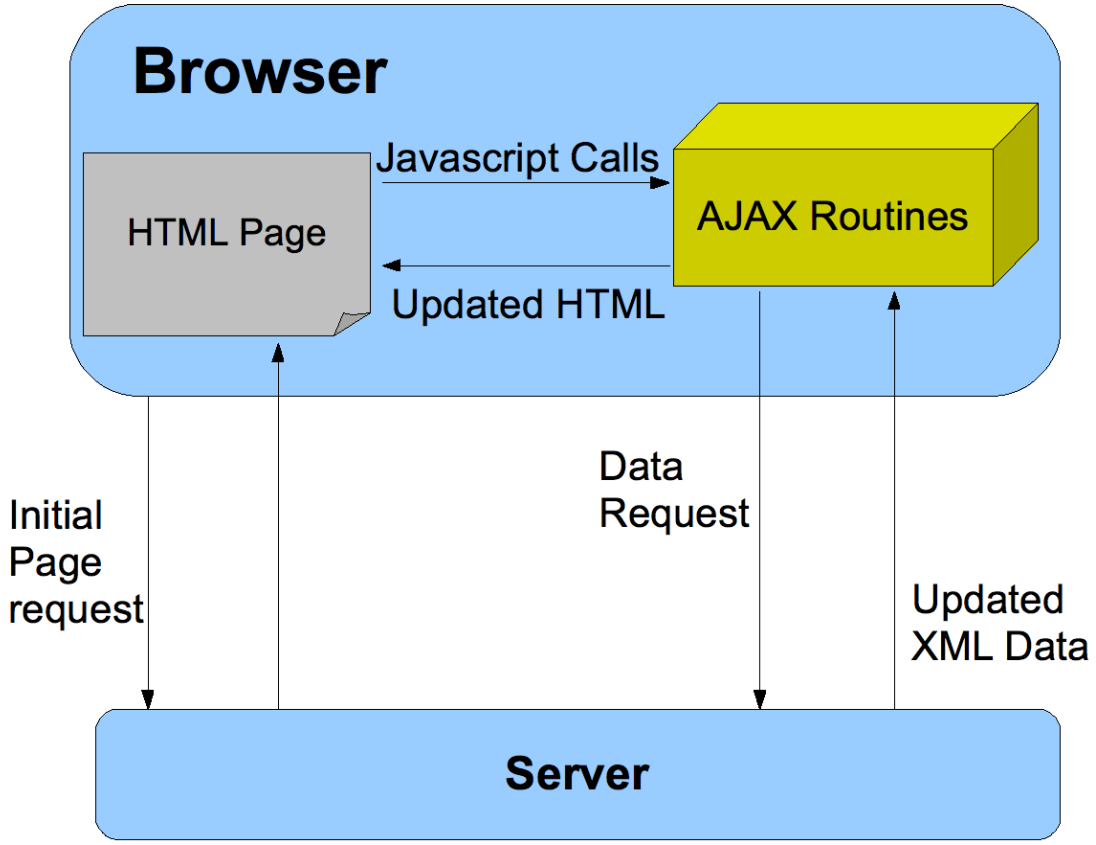

Figure 2 5: Ajax Operation

Chapter 3

Implementation and Design

In this section we comprehensively describe the approach we opted for implementing our home rental system. There are several ways for implementing CSP in the search problem domain as we have explained in the literature review section. In this section we explain the algorithms we choose over the other possible choices and also the in depth structure of the available features of our system. We have also described about the tools that we have used implementing our system. This part explains how CSP can make the system more dynamic rather the conventional way.

3.1 Implementation:

As shown in Figure 1, the tools has been used to built the architecture of the system is html (hypertext markup language), javascript, css (cascading style sheet), ajax, jquery, php and mysql. CSP algorithm and has been used in the search options.

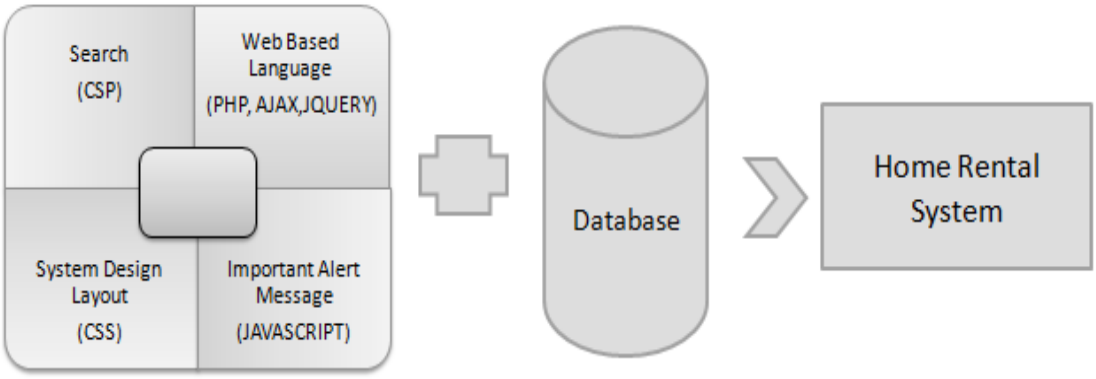

Figure 3 1: Architecture of the Home Rental System

We have implemented CSP algorithm in the search option instead of SQL search as CSP is more dynamic .In SQL search, foreign key and primary keys are maintained. Tthrough normalization process it creates 1st normal form , 2nd normal form and then 3rd normal form to give the desired domain result. [3] Working with SQL search will never give dynamic result. In SQL search, values are being processed through few steps manually. To make this home rental system dynamic we chose CSP. Therefore, tenants will prefer houses according to their priority, for that purpose a more dynamic search is must.

The target we are aiming to achieve here is to make the conventional searching strategy more dynamic by opting for an intelligent system. We have chosen Constraint Satisfaction Problem (CSP) algorithm. Now, there are six basic tree based search approach namely, naive backtracking (BT), backjumping (BJ), conflict-directed backjumping (CBJ), backmarking (BM), and forward checking (FC),Maintaining ARC Consistency(MAC). [1] We have chosen the problem domain Home-rental System by accumulating an additional feature transport tracking. We opted to choose CSP to implement the search feature in our designed system.

We have used the BT algorithm based search process for implementing CSP.

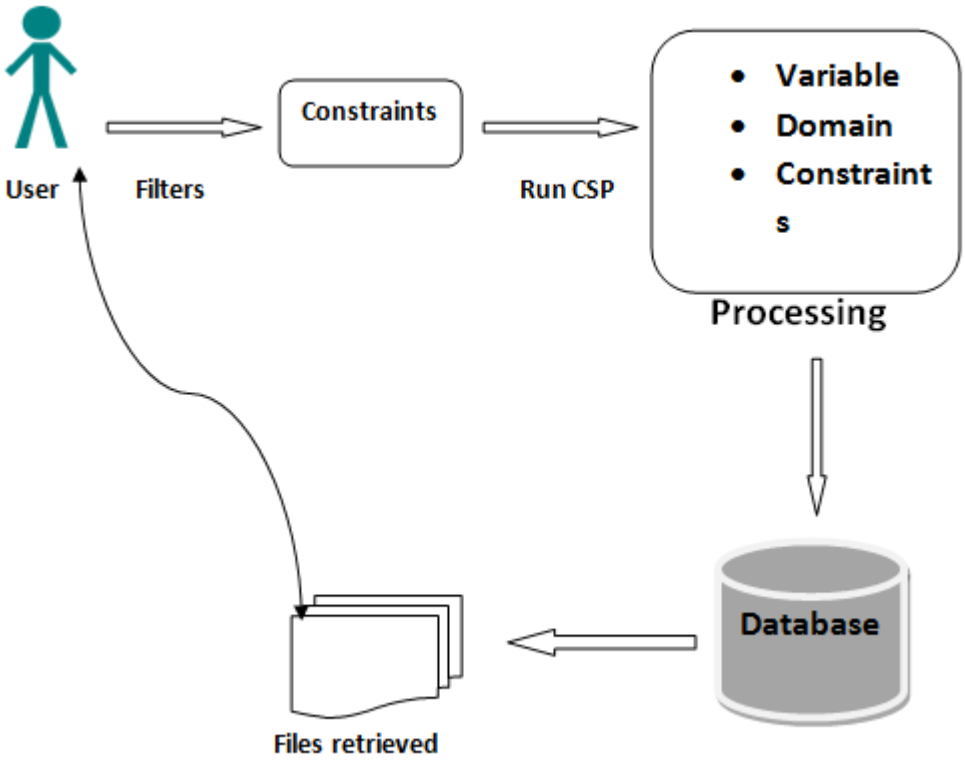

Figure 3 2: Basic Structure of CSP algorithm

We have tried to achieve the following features in our select problem domain of designing a modern dynamic home rental application-

- Dynamic Rental Search
- Rental posting
- Transport tracking
- Chat server
- Wish list
- Automatic mail alert system
- Google map view
- A responsive application of the system web view

3.1.1 Dynamic Rental Search:

- Step 1: User will be give a priority choice to select the most preferable variable which be the determination factor for creating domains. After setting priority user needs to set his desired rental preference in the given fields.

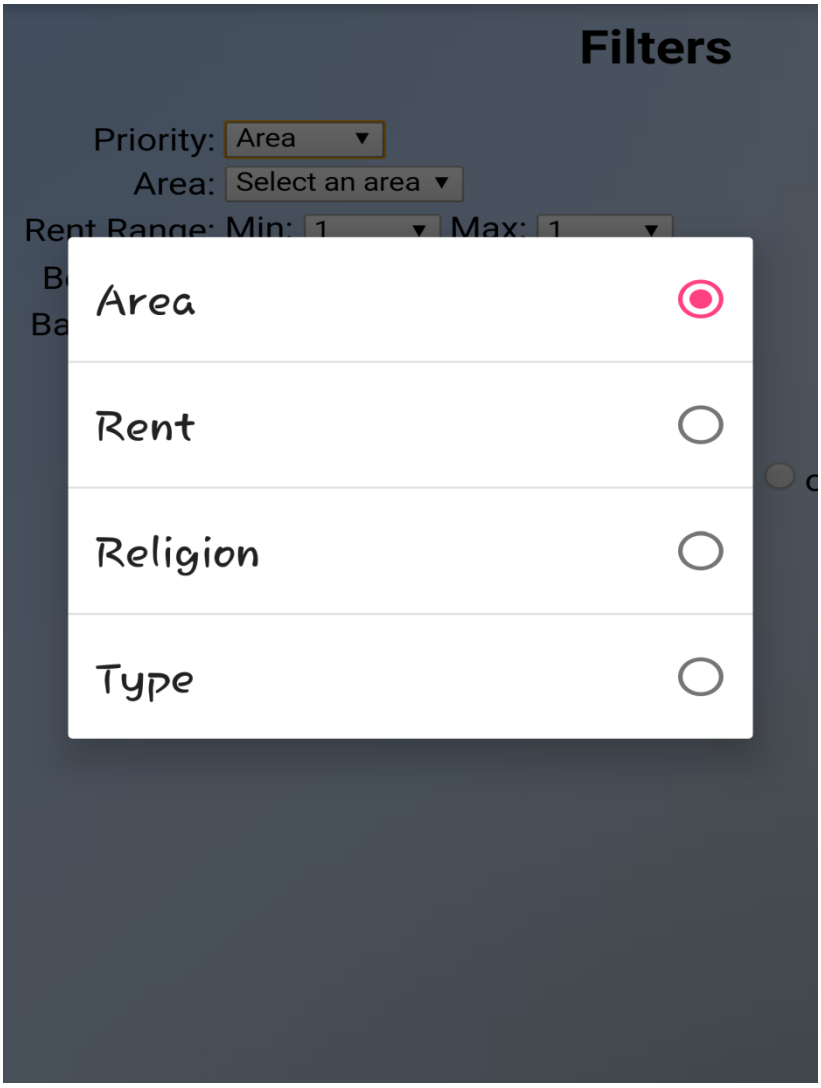

Figure 3 3: Setting the priority search

- Step 2: The values set in the field will be taken as the constraints of the algorithm. Depending on the priority selection the first domain will be created (i.e. if the area is priority then the resulting domain will be created based on the value user given in the area field by taking the value as constraint.) Solution is built up searching through space of the assignment of value to the variables. If a condition is falsified we immediately reject any possible ways to extend the assignment.

  For example we check whether the priority selected by the user is area or not. If it is area then we are pulling all the data from the table where area is similar to the area set by the user creating the first domain. This is the domain CSP is going to work upon.

  We have chosen the conventional BT for implementing CSP.

```
1.  BT(Level)
2.      if all var assigned
3.          Print values of each variable
4.          Return or Exit(Return for more solution Exit for only one solution)
5.      v=GetUnassignedvar()
6.      var[Level]=v
7.      Assigned[v]=true
8.      for d=each member in Domain(v)
9.          Value[v]=d
10.         OK=true
11.         for each constraint c such that
12.                 v is a var of c
13.                 and all other var of c are assigned
14.             if c is not satisfied by current set of assignments
15.                 OK= False
16.         if (OK)
17.             BT(Level+1)
```

18. return

Algorithm 1: Backtracking Search for finding the domain of values consistent with constraint

You can call one function, csp.solve(problem), which takes an object. This object should have 4 properties (though two are optional):

1. variables: object that holds variable names and variable domains as key-value pairs.
2. constraints: an array of constraints where each element is a list of head node, tail node, and constraint function that takes in two values (one for head node and one for tail node) and returns true if the constraint is satisfied, and falseotherwise. The nodes must be the names of the keys in variables. For the states coloring problem, [["CA","OR",not_equal_function],["CA","NV", not_equal_function],...] would be a valid constraints array. **Note:** ["CA", "OR", not_equal_function] and ["OR","CA", not_equal_function] are *different* constraints. If you want the constraint to hold both ways, you **must** include both constraints.
3. cb: Optional callback function for visualization. Passed in an object with variable and their assignments as key-value pairs and an object with unassigned variables and their domains as key-value pairs.
4. timeStep: Time between calls to cb in milliseconds. Default is 1 millisecond.

csp.solve(problem) returns an object with variable names and assigned values. If the problem cannot be solved we return failure

- Step 3: Other sub domain will be created based on the value user set in the non priority fields which will eventually lead to the final result domain showing the best suited result for the user.

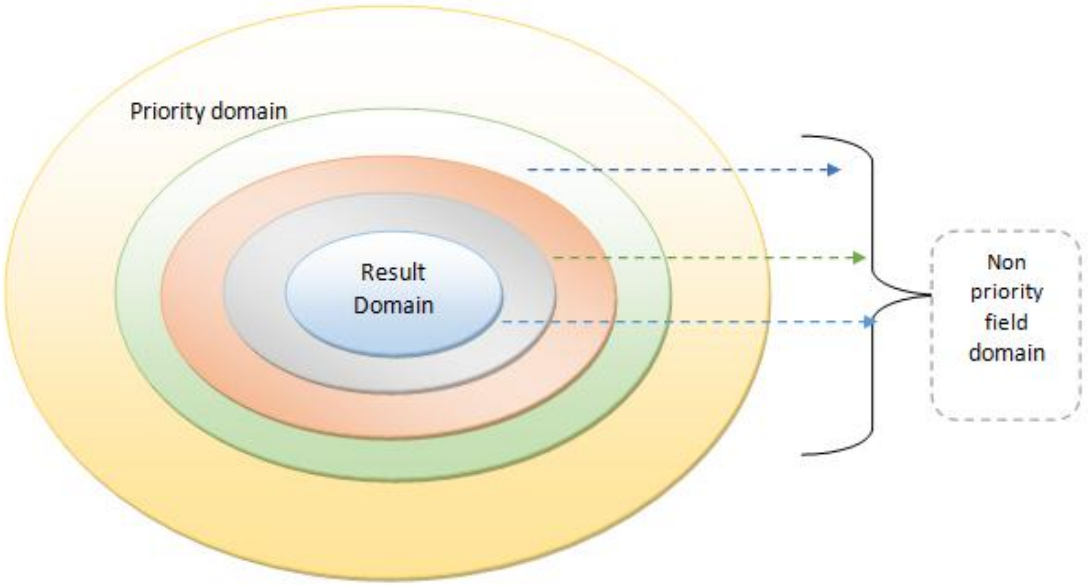

Figure 3 4: Basic domain creation strategy

3.1.2 Rental Posting:

Users will be provided a form similar to shown in fig 4.6 to post their houses for rental.

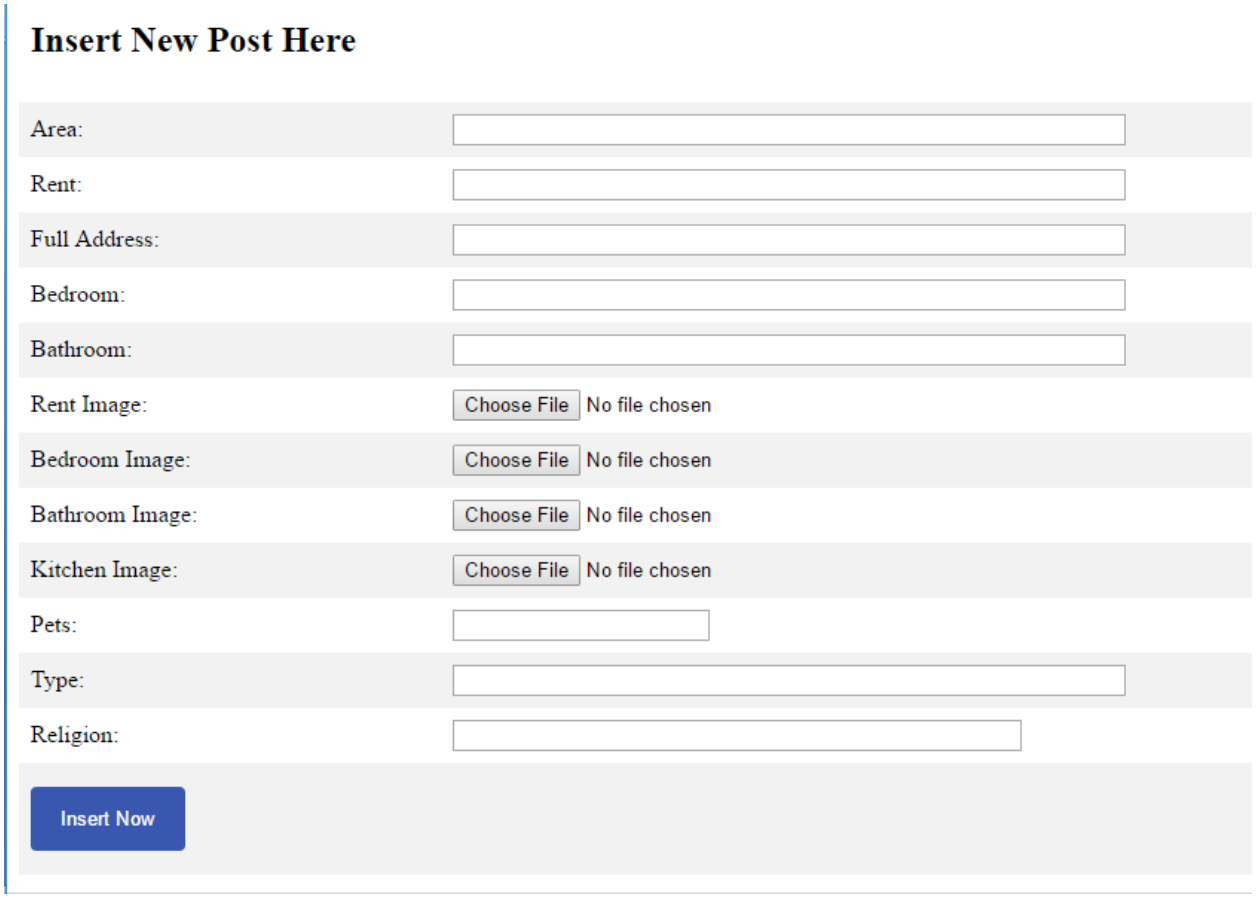

Figure 3 5: Rental posting form

The data inserted will be saved on the database.

3.1.3: Transport tracking:

The user can see the details of reaching his desired rental by searching for the available transport from his current location or any location he prefers by setting the location. the system will take

the user location and the rental location and will generate the available information of transport between these two point.

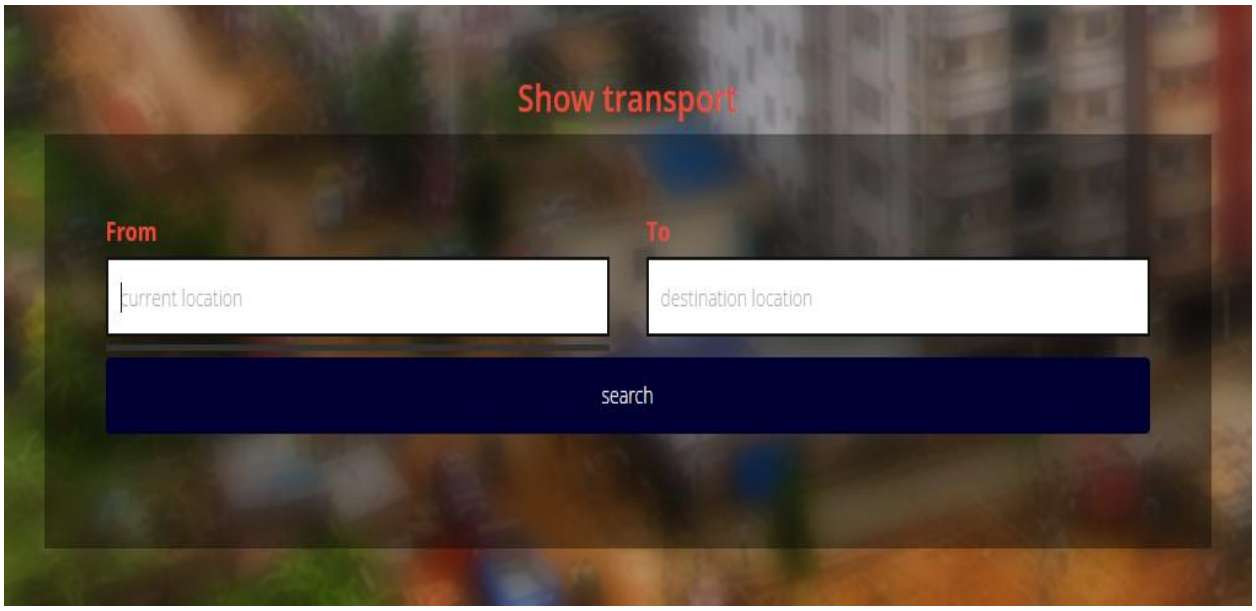

Figure 3 6: Web interface of transport searching

we take the input from user of source and destination and searching the database to track the transports available in these two routes.

3.1.4: Dynamic Mail Alert System:

The dynamic mail alert is implemented for informing the user if user has requested for a pad in his wish list. The system is built setting a triggering point in the rental insertion area therefore

whenever there is a rental posted the system will check whether is matches any wish list element. If it matches the user who requested the pad will be automatically sent a mail informing the availability of his requested rental. This approach is taken to take off the load of the system admin of manually mailing the user.

In our code will have the email address of the requester of the rental and we give the link to the rental. The php script that is triggered when we find a match while inserting a rental post in the database. We check the database of the wish list get information of theuser who requested it and send the mail.

3.1.5 An android based web-view:

We are provide our domain link to load through an apk file which will in return give us a web- view in our android phones.

3.2 Development Environment:

We have tested our programs in **XAMPP** version 3.2.2 which is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. It is incorporated with phpMyAdmin version 4.5.1. We have also hosted the website initially in a free webhosting site https://www.000webhost.com . We have used android studio for the purpose of creating a apk file to assist an webview of our system in android based phones.

3.3 System overview:

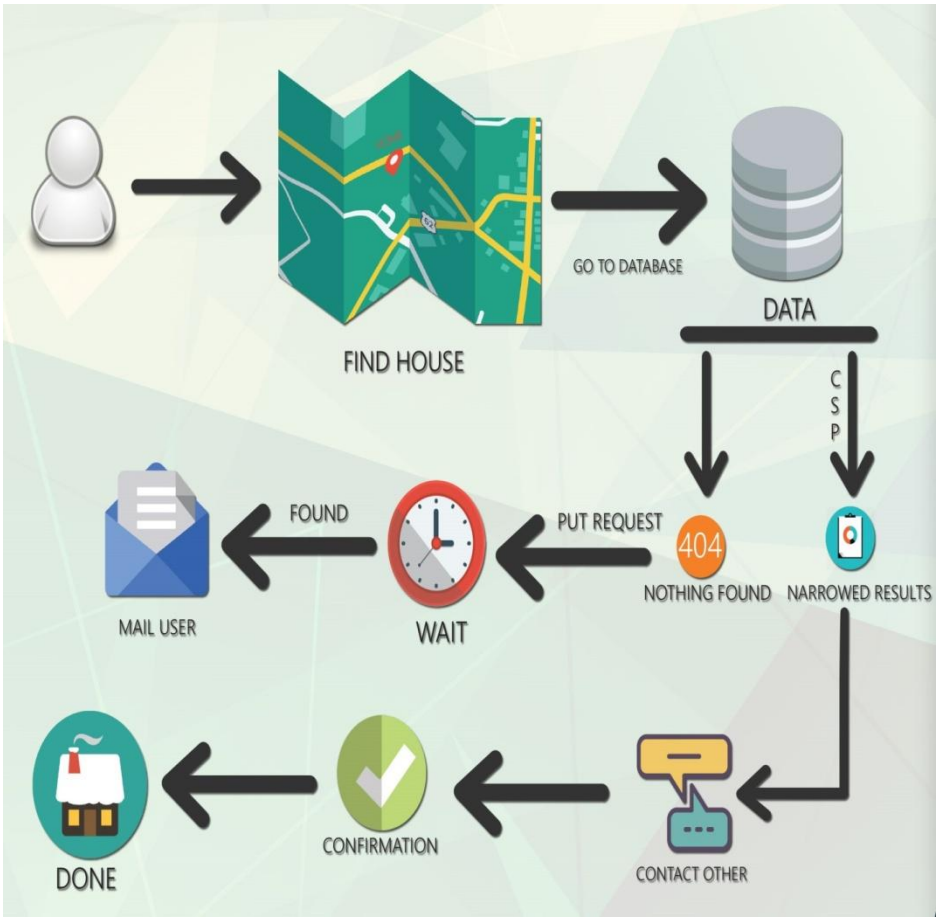

Figure 3 7: System Overview

Chapter 4

Result

In this result section we tried to offer graphical representation of the how system works against the given input. We tried to present few different results upon going for different input. What the user should expect from the system is stated in depth in this section. We have provided screen shots of how different features work of our designed home rental system. This section we provide a glimpse of the user interface of the system as well as how our implemented CSP algorithm functions for the search feature, transport tracking, dynamic mail alert approach.

4.1 Data Set:

This section describes the results of our implementation that were described in the previous sections. We tested the performance of our implementation using some manually inserted data. For initial stages we have used a few dummy data's for the purpose of doing the test needed for checking the algorithm is working fine.

4.2 User interface:

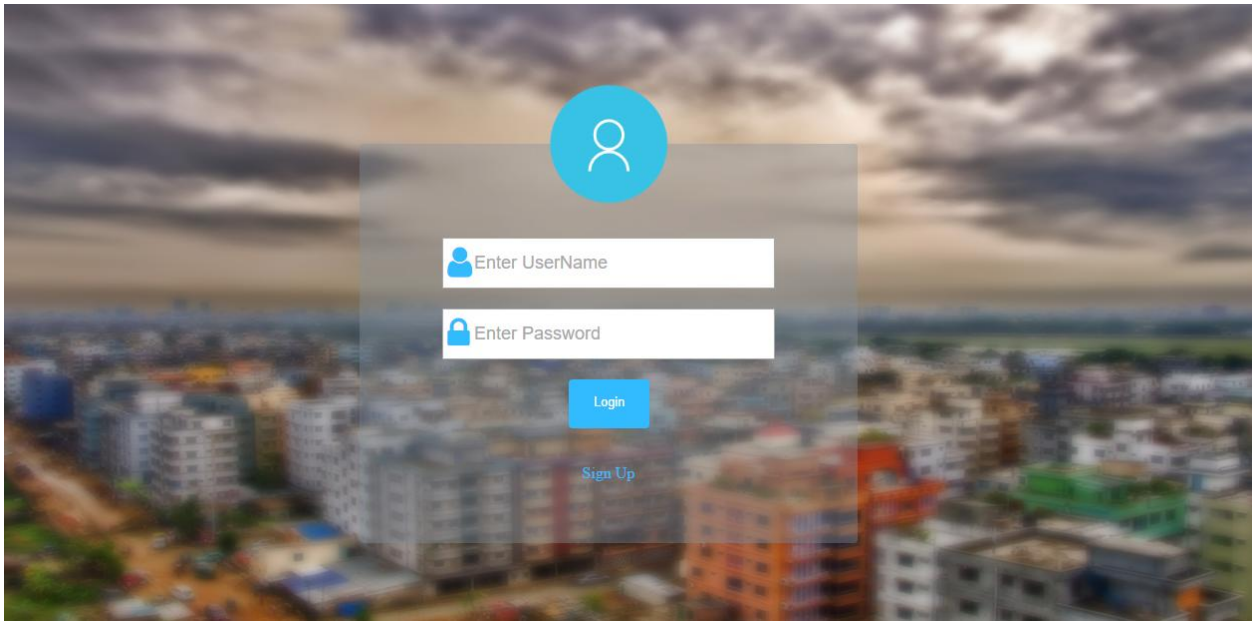

Figure 4 1: Login Interface

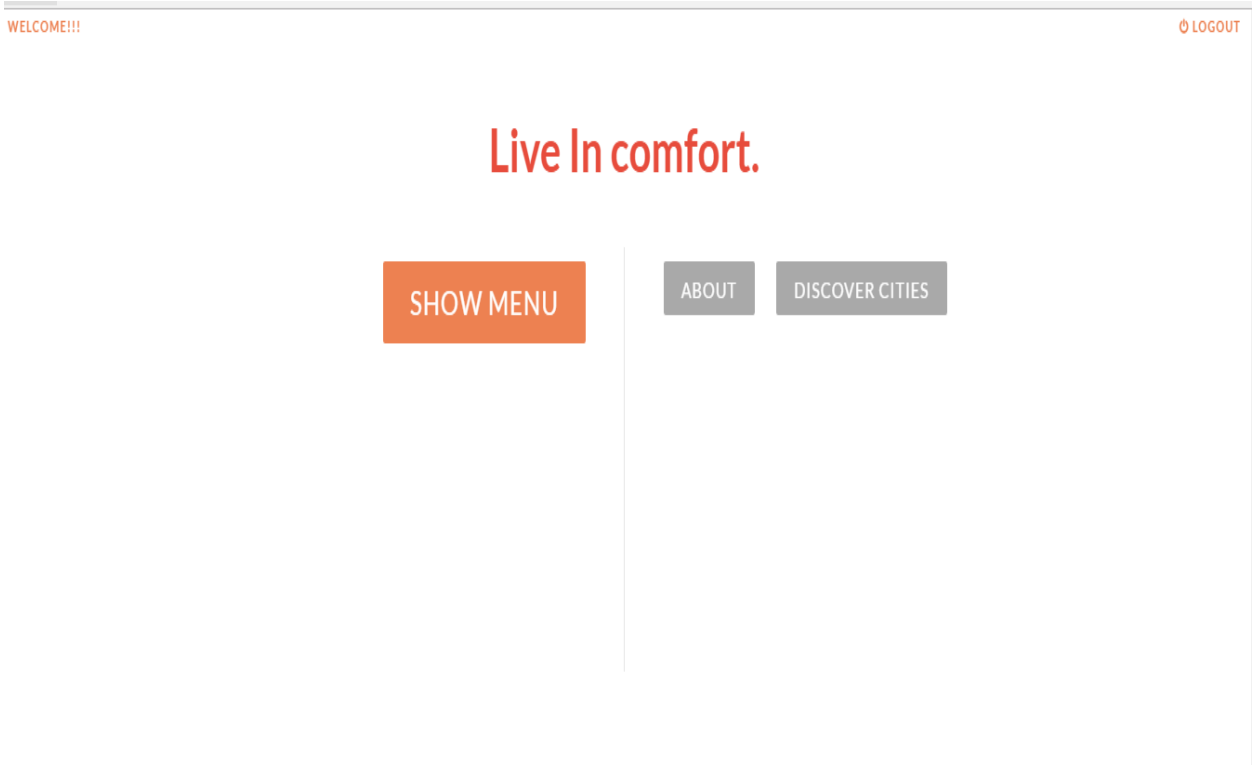

Figure 4 2: User interface upon logging in

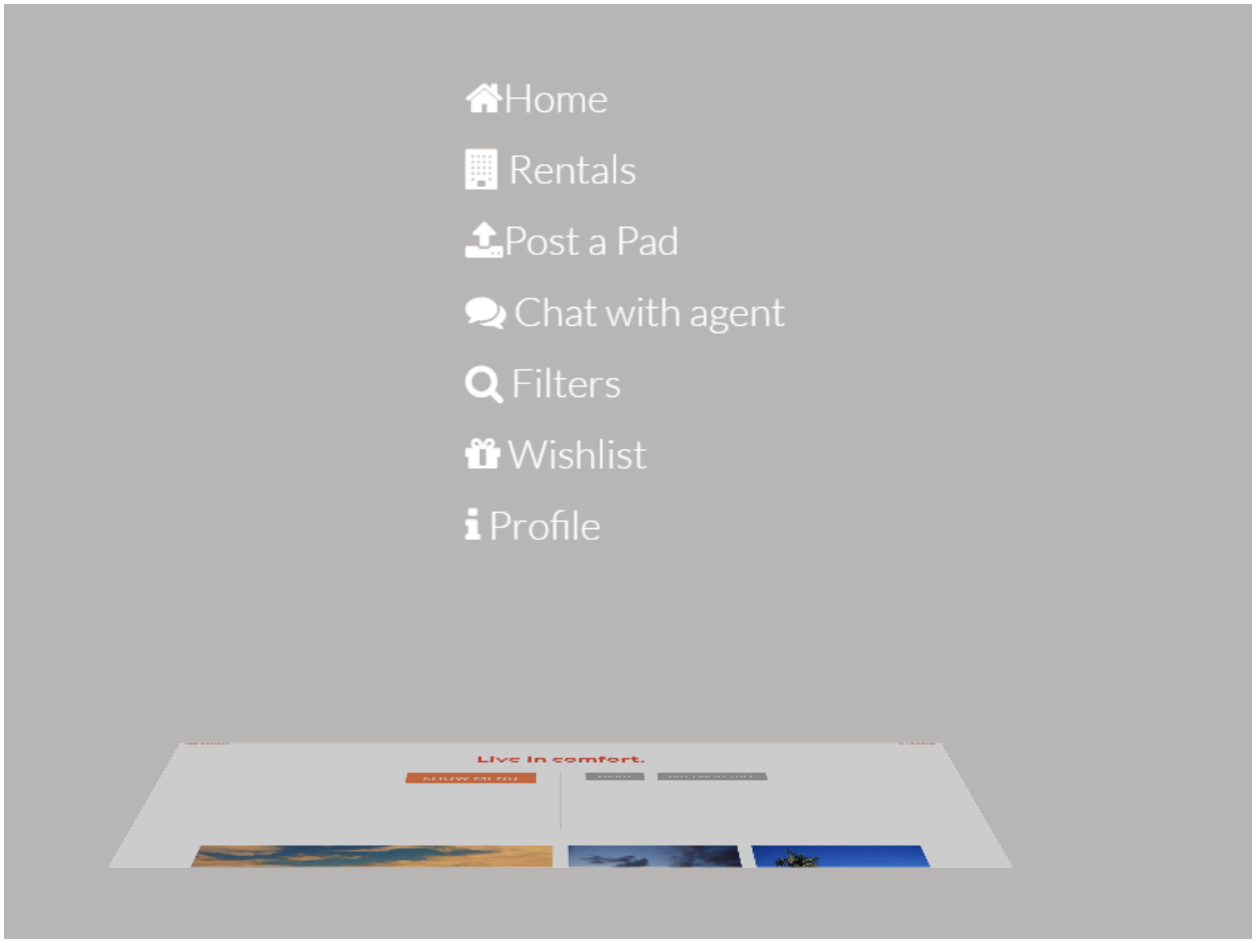

Figure 4 3: Menu Bar

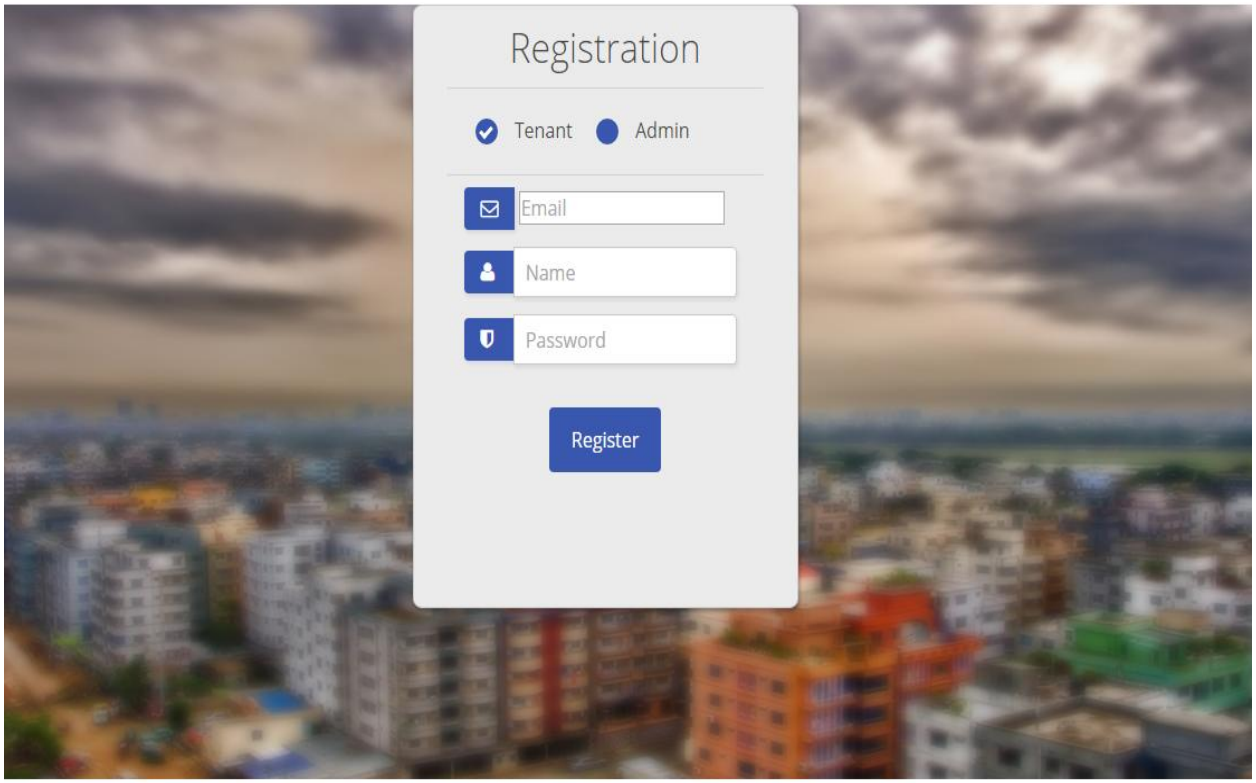

Figure 4 4: Registration Page

If the user is not yet a registered member he/she will be able to register providing the details mentioned in the figure 5.4. The user will not be directed to the homepage unless he or she signs in.

4.3 Results of Search feature:

As mentioned in the previous chapter of implementation we explained how one can opt for a choice of priority. Based on the selected priority the results will vary in some extent.

Figure 4 5: Search results based on Area set priority

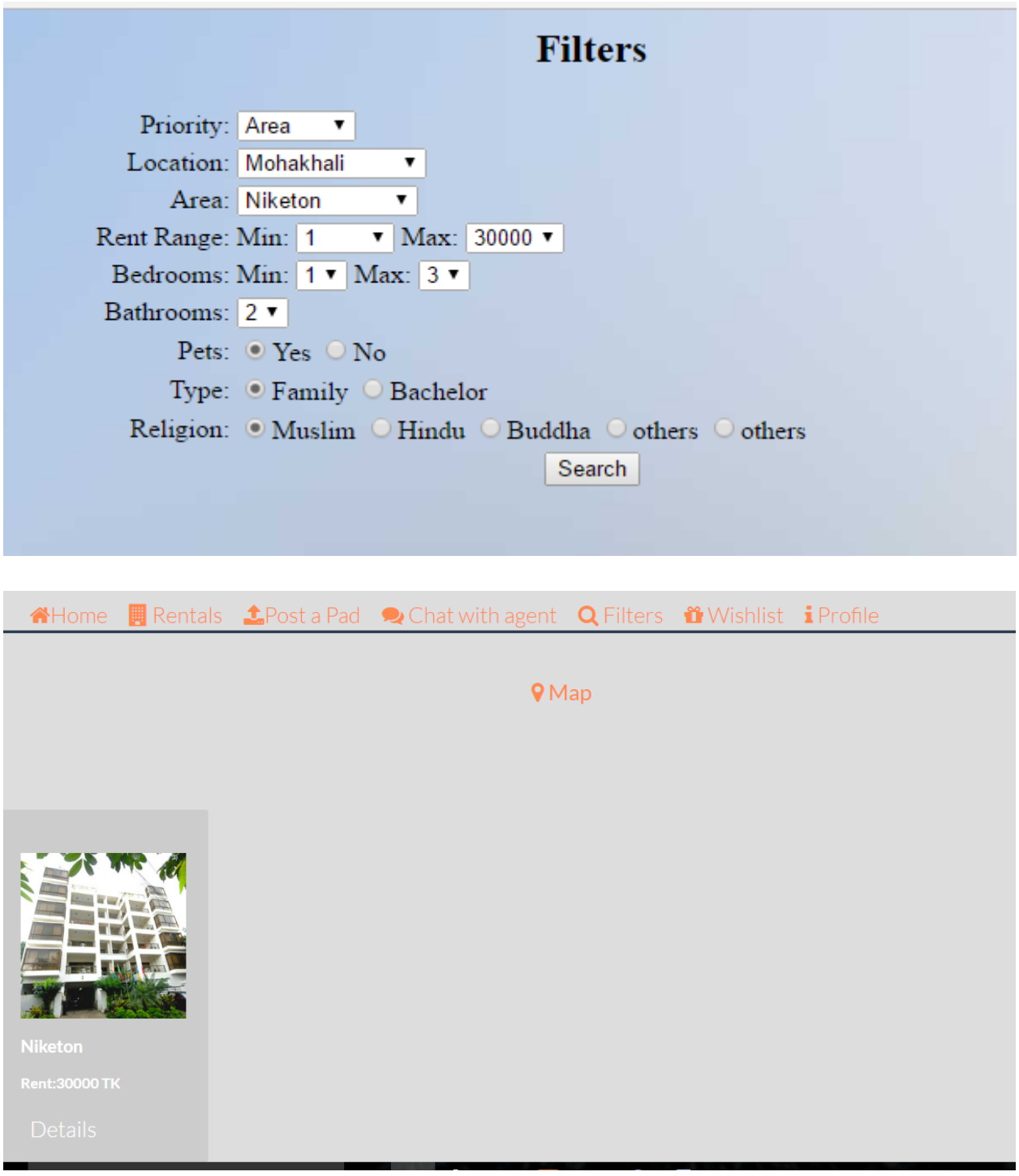

Figure 4 6: Search Results based on Rent set priority

The search results are incorporated with the Google map. Upon clicking the map button user will be able to see a visual of the location of the rentals through map. Not only the rentals, but the user can also find his location on the map.

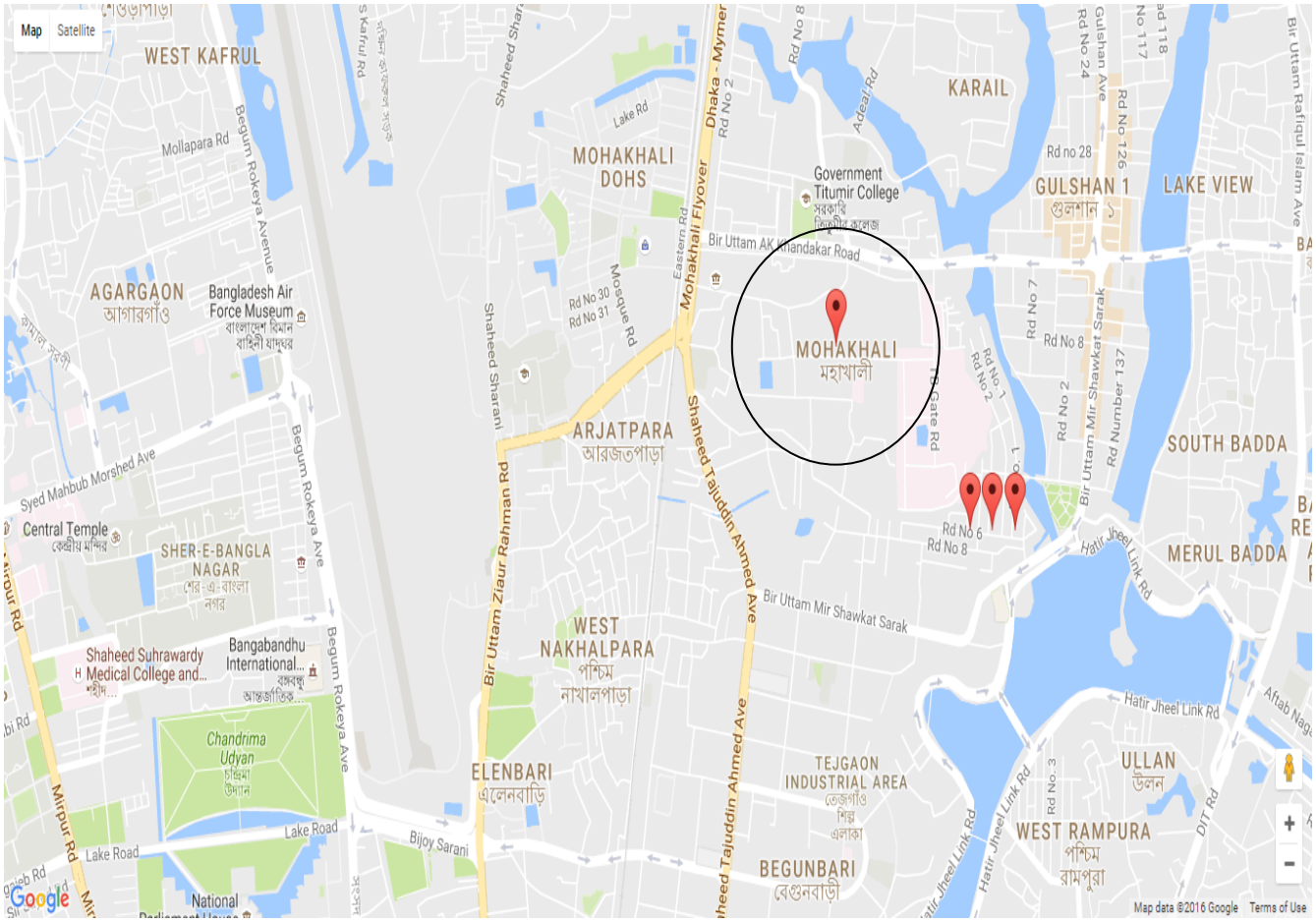

Figure 4 7: Google Map

The circled area shows the location of the user the user has provided the system while searching for rentals. Other the pointers show the results of the search based on area focusing the area Niketon.

Upon clicking the Details button attached with every rental shown in the box the user will be redirected to page where not only the user can see the details of the rental including pictures but the user can also search for the transport by manually providing the location as we have not integrated GPS system to automatically extract the user location yet.

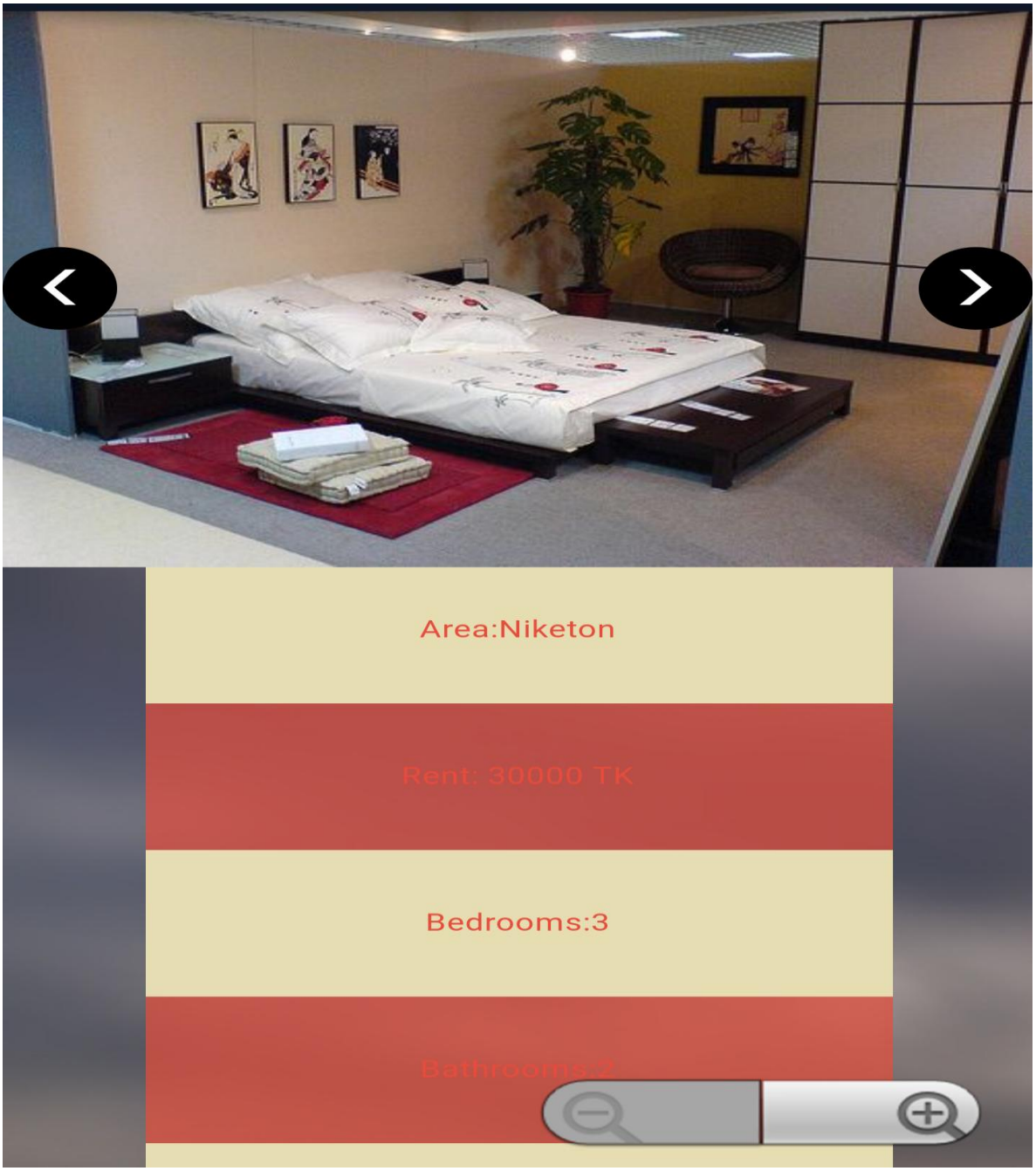

Figure 4 8: Details interface of a certain rental

User may browse through the pictures on the implemented slide show and also check the details entitled to the rental in further depth.

4.4 Mail alert:

Upon being uploaded any rental that matches any request existing in the wish list table will trigger a mail alert to the requester. User will receive a mail similar to the figure 5.9 notifying the user about the availability of the requested post.

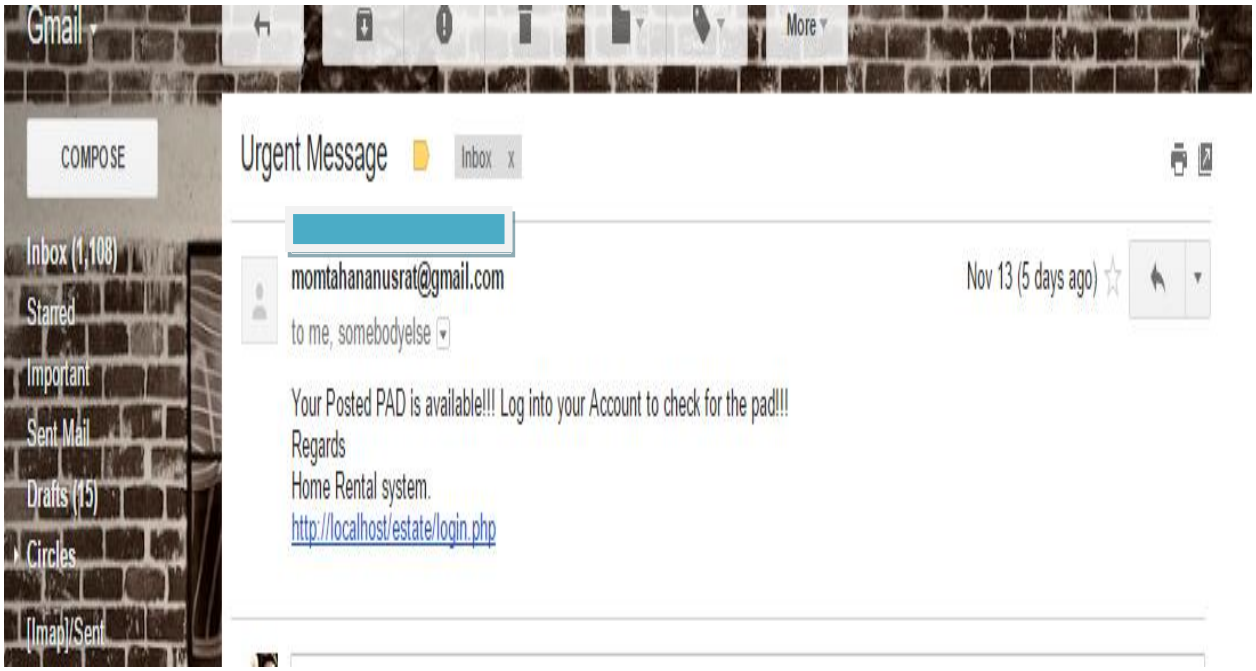

Figure 4 9: Mail Notification

Chapter 5

Conclusion and Future Work

In this following chapter, we sum our work in brief and conclude the paper. We also represent a future works which we could not able to do because of some contradictions we face during our research. We will work more on this and we have mentioned in the future work what we tend to o next.

5.1 Conclusion:

In this paper, we analyzed how the system can be made more dynamic. As the systems' main users are the general people so it has to be more user-friendly and dynamic. We made comparison with SQL search by implementing the search option in this. But, then we had to do that manually. In that case, general people would not be able to use that. Then we implemented CSP in the search option to make this dynamic. Finally, to give special concern on the case of renting house by bachelors, we implemented IOT to make that selection more precise and efficient. This work, to our best knowledge, is implementation of CSP algorithm in the search option of the system with a combination of transport tracker, dynamic mail alert and chat server.

5.2 Future Work:

In our thesis, we compared our work with SQL. But, there so many other algorithms which can be used in search field as well. Ths existing system will never reveal which algorithm they are using. Therefore, our next work is to compare it with the other algorithms and check for is it more dynamic than CSP or not.

Chapter 6

Reference

1. Gommans, H., Njiru, G. and Owange, A. (2014). Rental House Management System. *International Journal of Scientific and Research Publications*, [online] 4(11). Available at: http://s3.amazonaws.com/academia.edu.documents/35728176/ijsrp-p35101.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1480359917&Signature=e0UNdGLLuatBwTFr%2Bh16SiHRi50%3D&response-content-disposition=inline%3B%20filename%3DRental_House_Management_System.pdf [Accessed 28 Nov. 2016].

2. Rossi, F., Beek, P. and Walsh, T. (2006). *Handbook of Constraint Programming*. [online] Elsevier. Available at: http://www.informatik.uni-ulm.de/pm/fileadmin/pm/home/fruehwirth/Papers/FAI.pdf [Accessed 28 Nov. 2016].

3. Date, C. and Darwen, H. (1996). *A guide to the SQL standard*. 4th ed. Reading, Massachusetts [etc.]: Addison-Wesley.

4. Brailsford, S., Potts, C. and Smith, B. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3), pp.557-581.

5. Liu, Z. (1998). *Algorithms for Constraint Satisfaction Problems (CSPs)*. [online] Ontario: University of Waterloo. Available at: http://www.cs.toronto.edu/~fbacchus/Papers/liu.pdf [Accessed 29 Nov. 2016].

6. Mittal, S. and Falkenhainer, B. (1990). *Dynamic Constraint Satisfaction Problems*. Palo Alto: Association for the Advancement of Artificial Intelligence.

7. Barták, R., Salido, M. and Rossi, F. (2008). Constraint satisfaction techniques in planning and scheduling. *Journal of Intelligent Manufacturing*, 21(1), pp.5-15.

8. Joubert, N. (2012). *CSP.JS*. [online] GitHub. Available at: https://github.com/njoubert/csp.js/blob/master/README.md [Accessed 28 Nov. 2016].

9. Enqvist, O., Jiang, F. and Kahl, F. (2011). A brute-force algorithm for reconstructing a scene from two projections. *CVPR 2011*.

10. Bentley, J. (1979). Multidimensional Binary Search Trees in Database Applications. *IEEE Transactions on Software Engineering*, SE-5(4), pp.333-340.

11. Bessière, C., Meseguer, P., Freuder, E. and Larrosa, J. (2002). On forward checking for non-binary constraint satisfaction. *Artificial Intelligence*, [online] 141(1-2), pp.205-224. Available at: http://www.sciencedirect.com/science/article/pii/S0004370202002631 [Accessed 28 Nov. 2016].

12. Broder, A. (1990). Strategies for efficient incremental nearest neighbor search. *Pattern Recognition*, [online] 23(1-2), pp.171-178. Available at: http://www.sciencedirect.com/science/article/pii/003132039090057R [Accessed 28 Nov. 2016].

13. Koenig, S., Likhachev, M., Liu, Y. and Furcy, D. (2004). *Incremental Heuristic Search in Artificial Intelligence*. Georgia Institute of Technology.

14. Tutorials Point. (2016). *SQL RDBMS Databases*. [online] Available at: https://www.tutorialspoint.com/sql/sql-databases.htm [Accessed 28 Nov. 2016].

15. Tutorials Point. (2016). *What is AJAX?*. [online] Available at: https://www.tutorialspoint.com/ajax/what_is_ajax.htm [Accessed 28 Nov. 2016].

16. Kohonen, T and K Mäkisara. "The Self-Organizing Feature Maps". *Physica Scripta* 39.1 (1989): 168-172. Web.

17. Morton, A. and Mareels, I. (2000). An efficient brute-force solution to the network reconfiguration problem. *IEEE Transactions on Power Delivery*, [online] 15(3), pp.996-1000. Available at: http://ieeexplore.ieee.org/document/871365/ [Accessed 29 Nov. 2016].

18. Mackworth, A. and Freuder, E. (1985). The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, 25(1), pp.65-74.

19. Bacchus, F. (n.d.). *A Uniform View of Backtracking*. [online] Tortonto, Ontario: University of Toronto. Available at: http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.65.1771 [Accessed 29 Nov. 2016].

20. Shun-Zheng Yu, and Kobayashi, H. (2003). An efficient forward-backward algorithm for an explicit-duration hidden Markov model. *IEEE Signal Processing Letters*, 10(1), pp.11-14.

21. Bacchus, F. and Grove, A. (1995). On the Forward Checking Algorithm. In: *Principles and Practice of Constraint Programming - CP'95, First International Conference*.

22. Oddi, A. and Cesta, A. (2000). Incremental Forward Checking for the Disjunctive Temporal Problem. In: *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence*. [online] Available at: http://frontiersinai.com/ecai/ecai2000/pdf/p0108.pdf [Accessed 29 Nov. 2016].

23. Wallace, R., Grimes, D. and Freuder, E. (2009). Solving Dynamic Constraint Satisfaction Problems by Identifying Stable Features. In: *21st International Joint Conference on Artificial Intelligence*.

24. Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM*, 53(4), pp.10-11.

25. Ng, M. (2006). *SQL injection protection by variable normalization*. US20060212438 A1.

26. Oppel, A. (2009). *Databases A Beginner's Guide*. 1st ed. New York: McGraw-Hill Inc.

27. Greenspan, J. and Bulger, B. (2001). *MySQL/PHP Database Applications*. 1st ed. [ebook] Foster City: M&T Books. Available at: http://bootski.net/books/Technical.Ebooks.Pack.1/HungryMinds%20MySQL%20PHP%20Database%20Applications.pdf [Accessed 29 Nov. 2016].