ONLINE MUNICIPAL ADMINISTRATION SYSTEM

CONTENT

1. INTRODUCTION

The project ONLINE MUNICIPAL ADMINISTRATION SYSTEM collects the taxes from the citizen of the country, and each citizen also has to pay the electricity and water bills. All these work happened at the office of the respective department. A person has to visit the office to know about the taxes pending tax and bills. Today, when there are too many tax offices are present, and the user doesn't have much the time to face the problem. So to overcome this issue, we will design and application through which a citizen can know and pay the taxes and bills .The system will be based on the web to provide access to software from any locations.

Customer can login to our web application and search for their certificates like birth certificate, death certificate , marriage certificate and taxes based on location and select from list then they can search certificates which they want. Payment is based on types of certificates it maybe cash on delivery or some other plan. Ordered certificates are devilered to delivery address we entered at the time of purchase by municipality's staff. Our system help to users for purchasing certificates without going any municipality.

There are three different modules in our project they are admin, user and municipality. The user can login with their own unique username and password if they are registered. The system will display all the certificates a customer wants to purchase. Any modification to be done only by admin. He is also has the right to delete any items. Municipality can add all the certificates that the user wants. The certificates are birth certificate, marriage certificate and death certificate.

2. SOFTWARE REQUIREMENT SPECIFICATION

A software requirement specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements (Non-functional requirements impose constraints on the design or implementation such as performance engineering requirements, quality standards, or design constraints). The specification may include a set of use cases that describe interactions the user will have with the software.

Software requirement specification establishes the basis for agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by marketing and development division) on what the software product is to do as well as what it is not expected to do. Software requirement specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks and schedules.

The software requirement specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements, we need to have clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

2.1 SYSTEM REQUIREMENTS

Hardware requirements:

Hardware is a set of physical components, which performs the functions of applying appropriate, predefined instructions. In other words, one can say that electronic and mechanical parts of computer constitute hardware.

This project is built on a powerful programming language - PHP. It is a powerful server side scripting language. The backend is SQL Server, which is used to maintain database and the GUI is designed by using web technologies like html, css and javascript. It can run on almost all the popular microcomputers.

Software requirements:

The software is a set of procedures of coded information or a program which when fed into the computer hardware enables the computer to perform the various tasks. Software is like a current inside wire, which cannot be seen but its effect can be felt.

HARDWARE REQUIREMENTS

- PROCESSOR : Intel Pentium 4 /AMD Equivalent.
- RAM : 2GB
- HARD DISK DRIVE : At least 40GB
- KEYBOARD : Standard Multimedia Keyboard
- MOUSE : Standard 2 button Mouse
- MONITOR : Standard viewable color Monitor
- Internet Access
- Server OS

SOFTWARE REQUIREMENTS

OPERATING SYSTEM : Windows, Linux or Mac

FRONT END : PHP, CSS, JAVASCRIPT, HTML

BACK END : MYSQL

TOOLS : Text editor, WampServer /Xampp

3. SYSTEM ANALYSIS

The objectives of system analysis phase are establishment of requirements from system to be acquired, developed and installed. Analyzing the project to understand the intricacy form the vital part for system study. Problematic areas are identified and information is collected. Fact finding or gathering is essential to any analysis of requirement. It is necessary that the analyst familiarize himself with the objectives, activities, and functions of the organization in which the system is to be implemented.

3.1 EXISTING SYSTEM

The present Municipality administration system relies on the manual procedure and requires a man to maintain the public related information. The office contains the detail of tax of each person of the same city in the registers. Therefore, a citizen who wants to pay the tax has to visit the municipal office. The number of department of tax is huge, and a person has to visit the different office to know about the different tax which is nothing but the wasting time. The system is not user-friendly.

3.2 PROPOSED SYSTEM

The municipality administration system will centralize the tax department. This project will allow a user to know about the tax and it will give access to the function to pay those tax from the system. This system will automate all the work of the municipal department. It will update the information of each person and maintain the information. The system will be web based which will allow a citizen to know about the bills and tax from the home. Now they donot have to waste their time and can do other important work.

3.3 MODULES

Admin:

Admin login to his home page with his username and password. He can view municipality details, complaints and user approval. He can send add building taxes, roofing tax, area tax and add response.

User:

If the customer is the first time user of the system them he has to register his details to system and he will get his username and password to login to the system. After login he can view application , certificate, property, events. Here there is an application for birth, marriage, death, trade and trade renewal certificate. Here the user will apply for certificates by using the application .Then the municipality has to approve or reject. If the municipality approve application for certificate , the user can download the certificate. He is also able to change password etc .He is also able to pay for taxes like building , roofing and area tax . And also he can add complaint and view the response by the system.

Municipality:

If the municipality is the first time user of the system them he has to register his details to system and he will get his username and password to login to the system. After login he can view property. He has the right to approve or reject the property. Whenever the municipality approved their property the user must waits .Also we can search property using property number. He can add event. The municipality has the right to approve or reject the certificates like birth ,marriage ,death ,trade, trade renewal .He is also able to change password etc.

4. SYSTEM DESIGN

Based on the user requirement and the detailed analysis of the existing, the new system must be designed. This is the phase of the system designing. It is the most crucial phase in the developments of a system. The logical system design arrived at as a result of the systems analysis is converted into physical system design. Normally, the design proceeds in two stages:

- Preliminary or general design
- Structured or detailed design

Preliminary or general design: in preliminary or general design, the features of the new system are specified. The costs of implementing these features and the benefits to be derived are estimated. If the project is still considered to be feasible, we move to detailed design stage.

Structured or detailed design: in the detailed design stage, computer oriented work begins in earnest. At this stage, the design of the system becomes more structured. Structured design is a blueprint of a computer system solution to a given problem having the Computer Application same components and inter-relationships among the same components as the original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail. On the design stage, the programming language and the hardware and the software platform in which the new system will run are also decided.

4.1 INPUT DESIGN

Input design is the process of converting a user-oriented description of the inputs to a computer based business system into a programmer-oriented specification.

Inaccurate input data is the most common cause of data processing errors. If poor input design-particularly where operators must enter data from source documents-permits bad data to enter a computer system, the outputs produced are of little value. The input design process was initiated in the study phase where, as a part of the feasibilitystudy.

- Input data were found to be available for establishing and maintaining master transaction files for creating output records.
- The most suitable types of input media, for either off-line or on-line devices, were selected after a study of alternative data capture techniques.

In the design phase, as discussed earlier in this unit, the input design process was continued. Specifically:

- The decomposed data flow diagrams or expanded system flowchart identified master files (the database), transaction files, and the computer programs.
- The input media selected in the study phase were reviewed. Additional studies of alternatives were performed as required, and tasks were allocated among equipment, manual operations, and computer programs.

4.2 DATA FLOW DIAGRAM

The data flow diagram is also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design so it is the starting point of specification down to the lowest level of detail. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent the data flow in the system.

4.3 OUTPUT DESIGN

Computer output is the most important and direct source of information to the administrator. Efficient, intelligible output design should improve the system's relationships with the appraiser. A major form output, reports, is a hard copy from printer.

One of the most important features of an information system for users is the output it produces. Without quality output, the entire system may appear to be so unnecessary that users will avoid using it, possibly causing it to fail.

The term output applies to any information produced by an information system, whether printed or displayed. When analyst design computer output they

– Identify specific output that is needed to meet the information requirements.

- Select methods for presenting information.

Create document, report or other formats that contain information produced by the system.

5. SYSTEM TESTING AND IMPLEMENTATION

Before actually implementing the new system into operation, a test run of the system is done for removing the bugs, if any. It is an important phase of a successful system. After codifying the whole programs of the system, a test plan should be developed and run on a given set of test data. The output of the test run should match the expected results. Sometimes, system testing is considered a part of implementation process.

5.1 Types of testing

5.1.1 Unit Testing

Unit testing is the practice of testing small pieces of code, typically individual functions, alone and isolated. If your test uses some external resource, like the network or a database, it's not a unit test.

Unit tests should be fairly simple to write. A unit tests should essentially just give the function that's tested some inputs, and then check what the function outputs is correct. In practice this can vary, because if your code is poorly designed, writing unit tests can be difficult. Because of that, unit testing is the only testing method which also helps you write better code – that's hard to unit test usually has poor design.

In a sense, unit testing is the backbone. You can use unit tests to help design your code and keep it as a safety net when doing changes, and the same methods you use for unit testing are also applicable to the other types of testing. All the other test types are also constructed from similar pieces as unit tests, they are just more complex and less precise.

Unit tests are also great for preventing regressions – bugs that occur repeatedly. Many times there's been a particularly troublesome piece of code which just keeps breaking no matter how many times I fix it. By adding unit tests to check for those specific bugs, you can easily prevent situations like that. You can also use integration tests or functional tests for regression testing, but unit tests are much more useful because they are very specific, which makes it easy to pinpoint and then fix the problem.

When should you use unit testing? Ideally all the time, by applying test-driven development. A good set of unit tests do not only prevent bugs, but also improve your code design, and make sure you can later refactor your code without everything completely breaking apart**.**

5.1.2 Integration Testing

As the name suggests, in integration testing the idea is to test how parts of the system work together – the integration of the parts. Integration tests are similar to unit tests, but there's one big difference: while unit tests are isolated from other components, integration tests are not. For example, a unit test for database access code would not talk to a real database, but an integration test would.
Integration testing is mainly useful for situations where unit testing is not enough. Sometimes you need to have tests to verify that two separate systems – like a database and your app – work together correctly, and that calls for an integration test. As a result, when validating integration test results, you could for example validate a database related test by querying the database to check the database state is correct.

Integration tests are often slower than unit tests because of the added complexity. They also might need some set up or configuration, such as the setting up of a test database. This makes writing and maintaining them harder than unit tests, so you should focus on unit tests unless you absolutely need an integration test.

You should have fewer integration tests than unit tests. You should mainly use them if you need to test two separate systems together, or if a piece of code is too complex to unit test. But in the latter case, I would recommend fixing the code so it's easy to unit test instead.

5.1.3 System Testing

After the system is put together, system testing is performed. Here the system is tested against requirements to see if all the requirements are met and the system performance as specified by the requirements.

System Testing (ST) is a black box testing technique performed to evaluate the complete system, the system's compliance against specified requirements. In system testing, the functionalities of the system are tested from an end-to-end perspective.

System testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and non-functional testing

5.1.4 Acceptance Testing

Finally, acceptance testing is performed to demonstrate to the client, on the real life data of the operation of the system.

6. FUTURE ENHANCEMENT

This system is designed in such a way that , which is to be very useful for the any people who want their certificates from the authorized municipalities manages the details of the user by the convient database. Municipality administration system collects the taxes from the each citizen and generates the certificates without usage of paper works. The proposed system will allow the people to pay tax and get the certificates from the municipalities without wasting money and time. It has some advantages,

1. Easy to maintain.
2. Less time consuming

7. CONCLUSION

The Municipality administration system collects the taxes from the citizen of the country . All these work happen at the office of the respective department. A person has to visit the municipality for their certificates and taxes. Today, when there are too many tax offices are present, and the user does not have much time to face the problem. So to overcome this issue, we will design an application through which a citizen can know and pay the tax .The system will be based on the web to provide access to software from any location**.**

8. REFERENCE

- SQL Cookbook-“Anthony Molinaro”
- PHP & MYSQL Web Development
- www.php tutorial point
- www.w3layouts.com
- www.w3school.com