

# A Centralized administration system of rental shops for Municipal Corporation by using Angular Js Architecture

[1]K.Sudhakar, [2]M.Ramyasri, [3]MD.Reshma Sulthana, [4]T.Prasanna Kumari, [5]M.Sumanjali  
 [1]Associate Professor of CSE, [2,3,4,5] Project Student of CSE,  
 [1][ksudhakar@pscmr.ac.in](mailto:ksudhakar@pscmr.ac.in), [sudhamtech@gmail.com](mailto:sudhamtech@gmail.com), [2] [muppatiramyasri568@gmail.com](mailto:muppatiramyasri568@gmail.com),  
 [3] [mdreshmasulthana563@gmail.com](mailto:mdreshmasulthana563@gmail.com), [4] [tpasannakumari93@gmail.com](mailto:tpasannakumari93@gmail.com), [5] [anjalisumacse@gmail.com](mailto:anjalisumacse@gmail.com)

**Abstract**— any public or private agencies may face typical problems in maintaining their rental properties. Especially in the public domain of Corporations, Municipalities Towns and growing villages the publicly rented property is always under the people threat. In order to avoid legal and illegal obligations in the society a prospective model is needed so that the automation through such process may solve many issues of rentals in any business properties. In this application first we search for the rental shops of Municipal Corporation. if any rental shop available then we go for bidding .in that bidding process those who will pay more money for that rental shop ,those will maintain that shop for a particular period of time .if that given time period is completed then proceed for bidding.

**Index Terms**—Agile Process, Angular Js, Distributed model, relational database, payment gateway, messaging.

## I. INTRODUCTION

The rental system is under the subscription model of various business models available in the booming ecommerce market. The customer always is the prime focus of this environment. In this scenario the rental shops and renting blocks managed by the government organizations has become the typical task to initiate, process, execute and deploy for the society. <sup>1</sup>The municipal corporations approach to this is totally water fall model from <sup>2</sup>software engineering approach.

This paper suggests following the agile methodology of the project by following scrum. The municipal corporations are struggling to maintain the ledgers and paper works and the duplication has become a door window for illegal and irregularity for the administering tasks. The current scenario is explained as shown below:



Figure: 1.1 <sup>3</sup>Rental systems existing process in Municipal Corporations

## II. PROPOSED

### SYSTEM A. The base model:

In order to reduce the huge amount of paper work the model of the agile methodology helps a lot to come to a huge automating process of this system. The following diagram represents the flow of information based on Angular Js architecture model. The basic model from <sup>5</sup>Angular Js is as shown below:

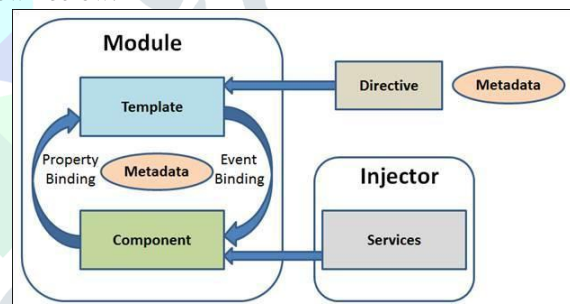


Figure 1.2: Angular Js Architecture as a base for the proposed system

### B. Illustration of the model:

As represented in the above diagram the core modules of the system Module, Metadata, Injector follows scrum process to complete the transaction for any instruction. The module composed of template, component and a metadata. The template receives the instructions from the directives and processes it with components then returns back, the injector in turn provides the services of communication and services. The proposed model takes the core of the above mentioned model and follows the <sup>6</sup>agile methodology with scrum. With the help of this methodology the automation is easy enough to maintain the software with reporting, tracking, communicating and the social justice is possible in the business environment as it will be followed proper rules of the government. The following requirements are observed for this

paper in both software and hardware areas of the system.

The software requirements are

- Angular Js 7
- Node Js 7
- MySQL 5 Database
- HTML 5
- XAMPP 5.6
- Crypto agility algorithms for secure payment gateway
- TEXT LOCAL SMS Gateway

#### Hardware Configuration:

- 500 GB HDD
- 4 GB RAM
- LCD Monitor
- Intel 5<sup>th</sup> Generation Processors

### C. The Proposed Mode using Agile:

*What Is Agile Web Development?: Agile web development is actually a broad category of methodologies based on the principles outlined in the [Manifesto for Agile Software Development](#), which was compiled by a team of professional developers in 2001. Specific methods such as scrum and XP are considered agile although they existed before the manifesto was written. Through their combined experiences of working with other developers, the authors recognized the value of adaptive planning and collaboration between self-organizing, cross-functional teams. The goal is to allow for flexibility and provide **rapid and continuous improvement of software solutions**. Early delivery is also a key goal of agile development, which entails streamlining projects by eliminating time-sucking tasks.*

Because they've been around for about two decades, a lot of the concepts behind agile development seem intuitive to younger developers; however, agile was a radical departure from older ways of doing things. Studies on the efficiency of agile web development compared to traditional methods are mixed, but agile remains popular in the professional world in 2017.

#### D. Traditional Web Development vs Agile Web Development#

Because the internet has evolved so rapidly in the past few decades, it's easy to forget that the World Wide Web isn't even 30 years old yet. During the infancy stages of web development, designers grappled with the basic task of translating the types of information you'd find in books into a website. They used paper prototypes, wire frames and flow charts to illustrate ideas. Use-case scenarios and focus groups were a primary source of feedback. Back then, CEOs would have their secretaries print out emails and then dictate responses, so there were obviously **large gaps of understanding** between users, developers and executives.

Developers soon discovered that creating software shouldn't be an entirely sequential process. There are always unexpected bugs as well as new technological hurdles to overcome, and anticipating the demands of users has become a science in itself. Fortunately, now that everyone is so well connected, it's easy to collect user feedback in real-time, so we have a greater understanding of how people interact with web applications.

The increase in internet users has coincided with advancements in content management systems, which has made it possible for anyone to design and edit a basic website with little training or expertise. These trends have given rise to industries that simply didn't exist a few years ago such as e-commerce. Since the process of making websites has become much more streamlined, developers have shifted focus to perfecting their methods to **address the growing needs of businesses and consumers**.

#### E. The Move to Agile#

The image below visualizes the traditional process of web development:

Source: [Sixrevisions](#)

As you can see, parts of the development process overlap, but there is a clear sequence of steps. Agile development, on the other hand, views most steps of development as ongoing and simultaneous:

Source: [Sixrevisions](#)

The biggest difference is that programmers are involved with content and navigation from the beginning. Individual challenges are addressed **as they arise** rather than waiting to make large-scale changes before launch. Beta launches have also become standard in order to start collecting feedback as early as possible, and maintenance is viewed as equally important to all of the other steps.

Therefore, everyone involved with a project from the top decision-makers to content creators to IT specialists should meet in-person during the early stages of development so that everyone agrees upon the overall goals. Though it may seem tedious at first, having everyone on the same page from the start saves time by reducing the need for constant emails, phone conversations, and meetings throughout the development cycle. Because all members of the team are engaged right away, what once took 4-8 months to finish can now be accomplished in just a few weeks.

#### *F. Agile vs Waterfall#*

The typical agile web development process is composed of a series of “sprints,” or cycles that involve discovery, design, development and testing. Every sprint results in a tangible product and new information to guide the next sprint. To better understand agile methods, it may be useful to compare it with the waterfall model, illustrated in the diagram below:

Source: [Tutsplus](#)

The waterfall model follows the traditional principles of sequential development whereas agile processes can be better visualized as a positive feedback loop:

Source: [Tutsplus](#)

Advocates of agile development may argue that the waterfall model is too rigid and ineffective; nonetheless, waterfall remains the norm in many workplaces, which is further evidence for why professional developers need to be adaptive to stay employed.

Aside from getting products to market faster, another advantage of agile methods is that they can require incremental investments. Then again, some investors are wary



of such arrangements because they don't fully understand agile development, which may be one reason why the waterfall model persists.

#### G. Agile Development Components#

An agile workflow usually consists of multiple components. The following provides a breakdown of the components found in a typical agile web development workflow:

- **Product Backlog** - A spreadsheet, chart or simply a physical wall covered in index cards that lists all of the features to be included in the final product. The product backlog provides a visual representation of the team's progress.
- **Sprint Backlog** - A list of the tasks to accomplish during a sprint. Before a sprint, the team chooses items from the product backlog and determines the tasks necessary to meet the user needs. Designers and developers independently assign priority and time estimates for each task, and that data is analyzed after each sprint to inform the next sprint. Some teams prefer to use a physical wall or whiteboard to keep track of tasks while others use free online task management tools like [Trello](#).
- **Scrum Meetings** - Short, daily meetings to check in and set the course for the workday.
- **Scrum Master** - The facilitator of scrum meetings tasked with managing communication between team members.
- **Shippable Increments** - Tangible, fully-functioning applications that showcase a feature or the final product. Ideally, each sprint will result in a shippable increment.

A quick tip that's worth mentioning as you populate your product backlog is to create [user personas](#). User personas are fictional characters that represent the goals, behaviors, and skills of your final product's potential users. Focusing on personas can help you identify **which features you need to include**. User personas may be derived from data collected through interviews or focus groups

#### H. Additional Benefits of Agile Web Development

Apart from what was already covered above, there are certain concrete benefits to using agile web development methodologies compared to traditional or other methods. The following section outlines these benefits.

##### 1) 1. Faster Feedback

First drafts are never perfect. In fact, sometimes you can follow a client's specifications to a T, yet the final product looks nothing like what they had in mind. By completing tasks in iterations and having something tangible to share with users after each step, teams can make changes along the way and **avoid a complete overhaul** six months down the line. Having brief, daily scrum meetings gives team members an opportunity to communicate about individual small issues and resolve them before they become big problems for everyone.

## 2) 2. *Keeping Up With Change*

The old adage “change is the only constant” couldn’t be truer when it comes to web development. Instead of viewing changes as unexpected obstacles to overcome, agile developers embrace change as an inevitable part of the learning process.

## 3) 3. *Higher Productivity*

Agile processes provide ample opportunities to measure productivity, which helps project managers better estimate workloads.

## 4) 4. *Lower Costs*

Faster and more efficient development eliminates the need for overtime pay.

## 5) 5. *Greater Customer Satisfaction*

Since applications go through so many checks before launch, customers are more likely to get a bug-free product that they are happy with.

## 6) 6. *Improved Worker Morale*

When team members get to see the fruits of their labor in shippable increments, they know they’re progressing in the right direction, which encourages them to keep working toward a goal. Team members are also more likely to feel a sense of shared ownership in the project, which is more motivating than artificial urgency.

## 7) 7. *Better Accountability*

Iterative methodologies make it easier to track the performance of individual team members, and it gives managers a way to measure the team’s overall commitment level throughout the development process. This information can help them set more realistic goals and give executives accurate timelines.

## 8) 8. *No More Detailed Project Plans*

Forgoing a project plan isn’t the same as not planning. It simply means that you don’t need one specific document that you update every time you learn something new. Agile development’s focus on frequent releases encourages an iterative learning process, so there’s no reason to waste time and energy managing a document.

## 9) 9. *More Collaboration*

In offices that still use the waterfall method, some team members may rarely interact. Designers often send off their designs to developers and simply move on to the next task. In agile workplaces, designers and developers often sit next to each other while working collaboratively on each iteration, which results in a more polished final product.

### I. *Keeping it Agile*

When trying to hold yourself or your teammates to agile principles, **ask the following 5 questions** whenever someone throws out an idea:

1. Will it help us achieve our goals?
2. Is it consistent with our brand?
3. Is there a better solution that already exists?
4. What is the worst that could happen if we tried it?

5. How will we evaluate whether or not the idea was effective?

Ongoing assessment is imperative to staying agile. Keeping what works and changing what doesn't helps everything move toward completion as quickly as possible.

#### J. Agile Web Development - In Summary

Web development is about much more than making websites and applications. Development also includes the business side of the equation, and programmers are constantly going back to tweak technologies to meet the specific needs of companies and their shareholders. A web developer's job is never truly ever done. Understanding the many facets of the development process will leave you better equipped to work in a professional environment. As agile web development becomes more and more widespread it's important to stay up to date on any new developments in the world of agile

development and practice it on a daily basis.

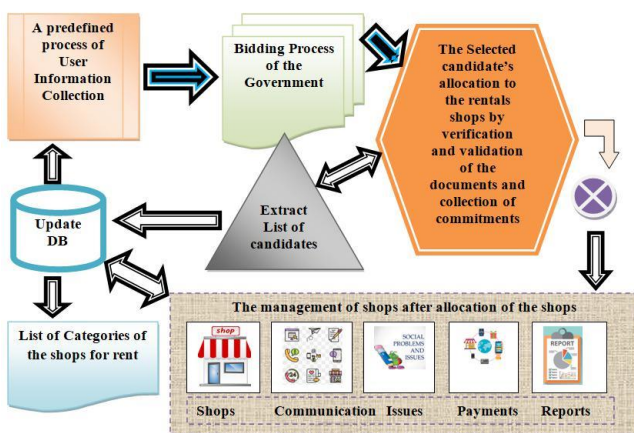


Figure 2.1: *The Municipal Shop Rental System*<sup>7</sup> architecture diagram.

### III. THE SOFTWARE ENGINEERING METHADODOLOGY-“CARS FOR MC”

For this project we have followed agile methodology to accomplish this task. The database is the scrum to report and respond back to the system every time since the inception of customer registration to the government. There are two other ways of collecting information as government may continue with its own ADHAR based information or collecting new information is a best process as it always contains the fresh data. The data is validated through email confirmation/ mobile OTP; then the registration is successful and confirms eligibility to gain the authority to lodge the application for bidding. Then bidding is organized in the designate date and venue so that the shops list is available and discussions can be accomplished accordingly. The following picture illustrates the basic approach of the core methodology we are using for these implementations.



Figure 3.1: *The basic agile approach for software engineering process*

**The algorithm:**

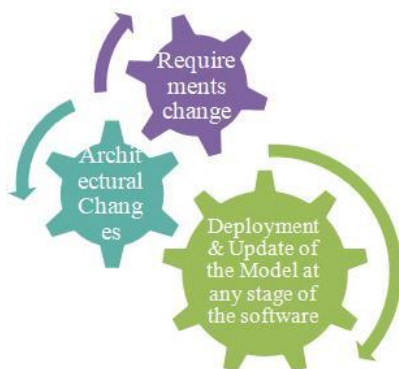
Algorithm for *Centralized Administration system of Rental Shops for Municipal Corporation*

```

Begin
Set ID, Customer Data, Credentials → DB DB ↔
Application Accepted
Set Accepted Application → Bidding Bidding
→ Selection of the Shop
Selected Shop → Commitment / Payment / leased agreement
Payment → Monthly / Other options Reports → Customer / Bank /
Government
Status → Communication [Messaging, Email, Physical
Verification]
End
    
```

**IV. THE FLOW DIAGRAM OF THE PROCESS**

The abstract view of the process is an incremental model with agile approach in both development and deployment stage. As the customer requirement increases the potentiality of the application increases and hence the result of the application stands without ambiguity in payment and reports process.



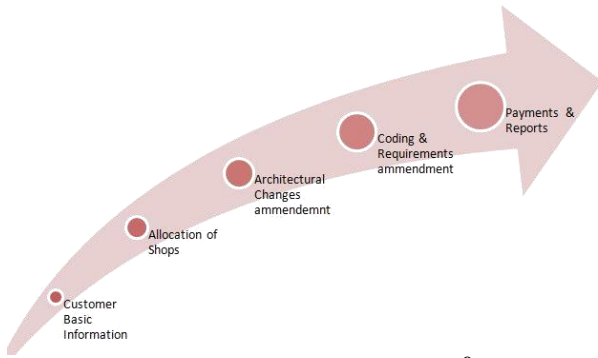


Figure 4.1: The agile process flow with<sup>8</sup> incremental approach

**9 Rational Diagrams:**

It is an object oriented model based for implementation of the code. We have used rational rose as design diagram platform. The various cases that are considered are illustrated in the form of diagrams.

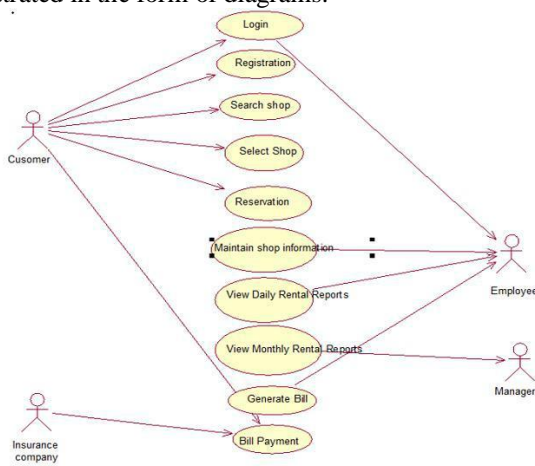


Figure 4.2: Use-Case diagram1 for "CARS for MC"

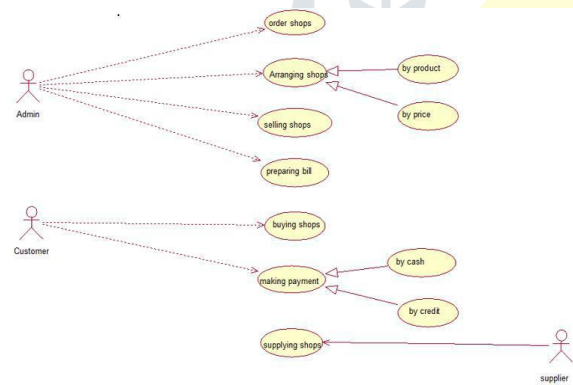


Figure 4.3: Use-Case Diagram 2 for "CARS for MC"

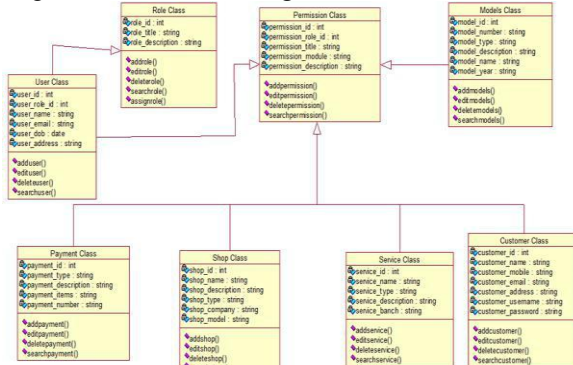


Figure 4.4: Class diagram for "CARS for MC"

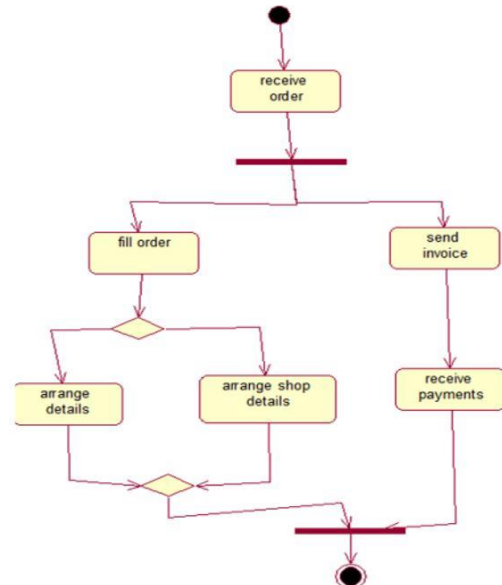


Figure 4.5: Activity Diagram for "CARS for MC"

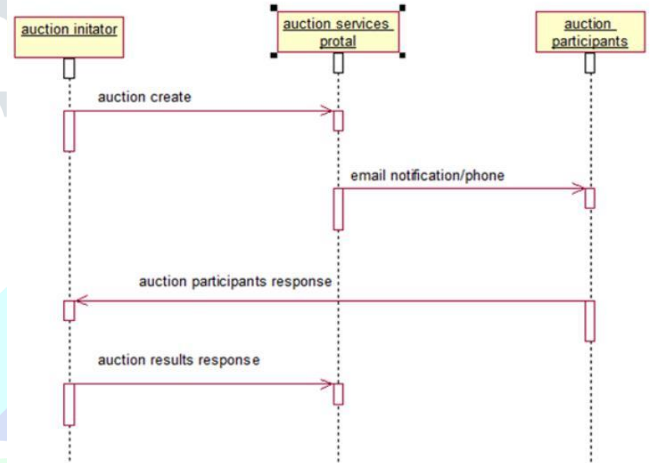


Figure 4.6: Sequence diagram for "CARS for MC"

**V. TEST CASES OF "CARS FOR MC"**

**A. Various Test cases under<sup>10</sup> Agile Testing Methodology applied to CARS for MC**

The behavioral driven approach, acceptance, exploratory testing cases are basically involved in this approach. This paper occupies all the test cases.

- a) *The behavioral driven testing is the one where the following steps are included and found for Boolean values.*
  - (1) Login and authentication
  - (2) Database functionality
  - (3) Messaging and communication
  - (4) Payment execution process
  - (5) Reporting validation, verification and delivery.
- b) *Acceptance Test case: Here in this paper the acceptance is validated as given below:*
  - (1) True cases of basic customer registration
  - (2) Approved cases of shops allocation
  - (3) Payment process results out of issues
  - (4) Valid and change drive reports
- c) *Exploratory Test cases are in situation where the amendments and changes that took place during the agile cases occurred in the cycle*
  - (1) Modification of the requirements
  - (2) Amendment of the features in architecture



(3) Final up gradation and release

The following diagram illustrates the basic testing methodologies of the agile process.

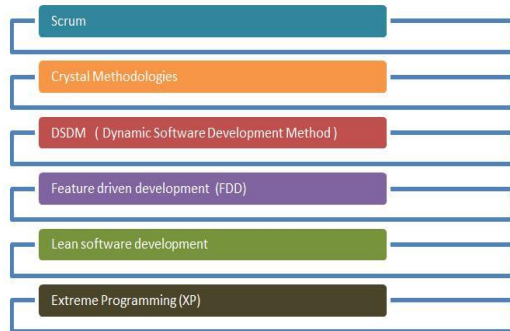


Figure 5.1: Agile Testing Process

## CONCLUSION

In this paper we have achieved the solution for the complete automation of the Rental shops for Municipal Corporation. This concept can be extended to distributed environment in both cloud and clusters using Hadoop and big data concepts. The usability of the mobile application can enhance the user interaction effective and easy for the tracking and monitoring of the system.

## REFERENCES

- [1] The times of India article on the Municipal Corporations issue in India. Link to study: <https://timesofindia.indiatimes.com/city/gurgaon/occupiers-of-corporation-shops-houses-get-a-chance-to-own-them/article-show/61759966.cms>
- [2] Software Engineering: A Practitioner's Approach By Roger S. Pressman pp 22
- [3] Optimization of Rental Systems: Queuing Loss Theory for the Optimization of ... By Felix Papier pp 6
- [4] AngularJS By Brad Green, Shyam Seshadri pp 11
- [5] Introduction to Agile Methods By Sondra Ashmore Ph.D., Kristin Runyan pp 15
- [6] Just Enough Software Architecture: A Risk-driven Approach By George Fairbanks PP 22
- [7] Effective Project Management: Traditional, Adaptive, Extreme By Robert K. Wysocki, Rudd McGary pp 17
- [8] Visual Modeling with Rational Rose 2002 and UML By Terry Quatrani 84
- [9] Visual Modeling with Rational Rose 2002 and UML By Terry Quatrani pp 102
- [10] Agile Testing: A Practical Guide For Testers And Agile Teams By Crispin pp 402

## BIOGRAPHY



**Mr.K.Sudhakar, Associate Professor of CSE.**  
Scholar at Andhra University, Research interests are Distributed Computing, Cloud, Big Data, Network Security, Information Systems.



Ms M RamyaSri, MD.Reshma Sulthana, T.Prasanna Kumari, M.Sumanjali

**Project Students Computer Science & Engineering**