

# Daily Needs Management Application

Akanksha Tyagi, Komal Gulhane, Lisha Mamarde, Neha Mokashi

Student, Dept. of Information Technology, YCCE Nagpur, Maharashtra, India

-----\*\*\*-----  
**Abstract** - Now-a-days every aspect of our lives is inclined towards technology. Technology is catering to people from every walk of life. From buying and selling second-hand things online to booking on-the-go taxi service, information technology has made everything hands-on. More recent examples include ordering medicines online. In such a world where almost everything has become tech-savvy, managing daily needs like milk, bread, newspaper, eggs etc. is still manual and quite hectic. These groceries are usually delivered to us in the morning. So it is a tedious job for people to manage these things simultaneously. To ease this job there should be a single platform where you can manage your daily needs and monthly bills too. So this project is about providing a single platform to user where he can set his default list of services and also can update according to his needs and can even keep track of his or her bills. Also, as the word 'management' comes to play, it becomes feasible for one to develop an application and use it anywhere. That means it is more feasible to develop once and deploy it on any platform i.e. operating system should not be a barrier. Therefore a relatively new technology called as 'Phonogap' can be used to develop hybrid application. The backend of this application is developed using NodeJS and the database is developed in MongoDB. The frontend is designed with the help of HTML 5, Ionic framework, Cordova and Angular.js.

**Key Words:** Hybrid application, daily needs, vendor, customer, services, delivery.

## 1. INTRODUCTION

India is a country which is driven by the people of the middle class. People, who have fixed monthly salaries, wound in a routine lifestyle. IT sector has made the lives of such people easier by leaps and bounds. Social networking sites have helped people to connect with one another whereas e-commerce has made shopping a less tiresome job. One cannot forget the revolution of on-the-go taxi cabs which have made travelling a better experience. Inspired by such daily use applications, we propose the idea of 'Daily Needs Management System' mobile application.

This paper focuses on the issues that people face in managing their everyday morning requirements. Consider the morning scenario in a common Indian

household where everybody is in a hurry to get ready for the day. Apart from getting a head start for your day, one has to make sure that there are enough basic supplies for your morning coffee or for making sandwiches for the children of the house and just hope that the newspaper arrives in time. Generally, the women of the house take these responsibilities; therefore, they manage it like clockwork. But it takes a toll on them also. This application intends to minimize the number of people rendering every morning services like milk, bread, eggs and newspaper. Only one delivery will take place every day at a predefined time with the deliverables as ordered the previous day. And the user may change his order as and when required and may even cancel the order for a specific day or a set of days if he/she is going out of town.

On the other end, vendors of these goods will have a wider customer base and consequently it becomes more profitable for them. The vendor selection will be done on a location basis. The customers will have the luxury to select the vendor of their choice. The logistics part will be handled by the vendor and his party will make sure that all deliveries are managed timely and efficiently. The bill thus produced at the vendor's end will be traceable by the customer and an invoice will be generated at the end of the month.

Therefore to manage all these functionalities, we propose a mobile application which provides one single platform where the only input provided by the customer will be the order for the next day and the deliverables will reach the customer in one visit.

## 2. EXISTING WORK AND PROPOSED SYSTEM

### 2.1 Working of Existing System:

The present scenario for daily services such as milk, eggs, newspaper, curd etc. is completely manual. Different doorstep deliveries are required for different goods for example we have to entertain milk and newspaper delivery boys separately which is cumbersome. Keeping an account of received goods is another task that a customer needs to do. Due to this, budgeting gets hampered because one may not remember whether he/she received goods on a particular day. Also when one shifts to a completely

unknown city or area, it becomes difficult for him to find trust-able vendors nearby.

The latest development to solve this issue is the new web application called Grofers. Grofers is an online delivery service that connects consumers with local grocery stores. One can get connected to nearby grocery store and can give their groceries (vegetables, Cosmetics, mobile and accessories etc.) order online without any hassle, also it allows one to schedule time of delivery as per his/her convenience. As this application does not work on daily basis that means you order your stuff and it gets delivered to you, what if one wants some services such as bread, canned milk, eggs etc. almost on a daily basis then they will have to go through the procedure of placing the order daily, this is tiresome.

#### *Disadvantages of Present Working System:*

- Not user friendly
- Difficulty in monthly budgeting
- Manual Control
- Multiple visits by different delivery boys
- Uncertain deliveries.
- Time consuming

These disadvantages will be overcome by our proposed system. The proposed system will be more user friendly because user will be able to handle his/her daily needs automatically with the help of this application. Also, there would be just one delivery which will be at comparatively certain time.

## **2.2 Working of Proposed System:**

In this paper, we have proposed a system which has two interfaces one for customer and one for vendor. In the vendor interface, after successful registration as a vendor, he/she will add the goods (milk, curd, newspaper etc.) that he/she is willing to provide. He/she must also specify the deadline time for taking new orders from customers. Vendor has the option of either accepting or rejecting request from customer for delivering goods to them. He/She will also get notified when a customer modifies his/her default order. Also vendor will be able to keep track of the payment of all of his customers.

On the customer end, after successful registration as a customer, he/she can select a vendor as per his/her needs and also according to his/her nearby location. Once the vendor accepts the request of the customer, the customer can choose goods according to his/her requirement and specify details like quantity, duration for which he requires the service and frequency of service etc. If for some reason, the customer wishes to

unsubscribe to any service, he/she can use the undo service option to do so. For ease of budget keeping this system also provides a functionality to track his/her current bill i.e. amount of money that the customer owes to the vendor till date. Apart from current bill he/she can also keep a track of the total estimated bill i.e. amount of money that the customer owes to the vendor till the end of subscription of all the services.

## **3. TECHNOLOGY USED FOR THE PROPOSED SYSTEM**

### **3.1 HTML 5**

HTML5 is a mark-up language used for providing structure and presentation to the content on the World Wide Web. It is the fifth revision of the standard from the date of the inception of the World Wide Web.

It aims to improve the language with support for the most current multimedia. It keeps the script easily readable by humans and efficiently understood by computer. HTML5 is intended to include HTML 4, XHTML 1 and DOM Level 2 HTML.

In the proposed system, HTML 5.0 is the language which has been used for designing the pages of the application.

### **3.2 Ionic Framework**

Ionic is a completely open-source software development kit for developing hybrid mobile applications. Ionic is created to be integrated with AngularJS and Apache Cordova. It provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Applications can be built with these Web technologies and then distributed on mobile devices by integrating with Cordova. Ionic was created by Max Lynch, Ben Sperry, and Adam Bradley of Drifty Co. in 2013.

In the proposed system, Ionic Framework has been used for developing the mobile application and for adding aesthetic value to the system.

### **3.3 AngularJS**

*AngularJS* is an open-source framework used for web applications maintained by Google and some other corporations to address many of the challenges encountered in development of single page applications. Its aim is to simplify the phase of development and testing of above mentioned applications by providing a framework for model-view-controller (MVC) and model-view-view-model (MVVM) architectures at the client side, along with components commonly used in web applications.

The working of AngularJS library commences by first reading the HTML page, which has embedded tag attributes. Angular interprets these tag attributes as directives for binding parts of the page to a model that is then represented by standard JavaScript variables. The values of the mentioned JavaScript variables can be manually set in the code, or retrieved from various JSON resources.

In the proposed system, AngularJS is used to add functionalities to the various buttons in the pages of the application.

### 3.4 MongoDB

MongoDB[1] is a cross-platform document-oriented database[2]. Classified as a NoSQL[3] database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like[4] documents with dynamic schemas, which makes the integration of data in applications faster and easier comparatively. MongoDB is developed by MongoDB Inc. It is published as free and open-source software. MongoDB is the fourth most popular type of database management system, and the most popular for document stores.

The database of the proposed system has two collections in MongoDB. One collection is for storing the data of the customers and the other collection is for storing the data of the vendors.

### 3.5 Express JS

Express JS is a web application server framework, designed for building single-page, multi-page, and hybrid web applications. It is the standard server framework for Node JS. The original author, TJ Holowaychuk, described it as a Sinatra-inspired server, meaning that it is relatively minimal with many features available as plug-ins.

Express is the back end aspect of the MEAN stack, together with MongoDB database and AngularJS front end framework.

In the proposed system, Express JS acts as the middleware between front end and back end. It processes the request which comes from the client side and sends appropriate response from the server side.

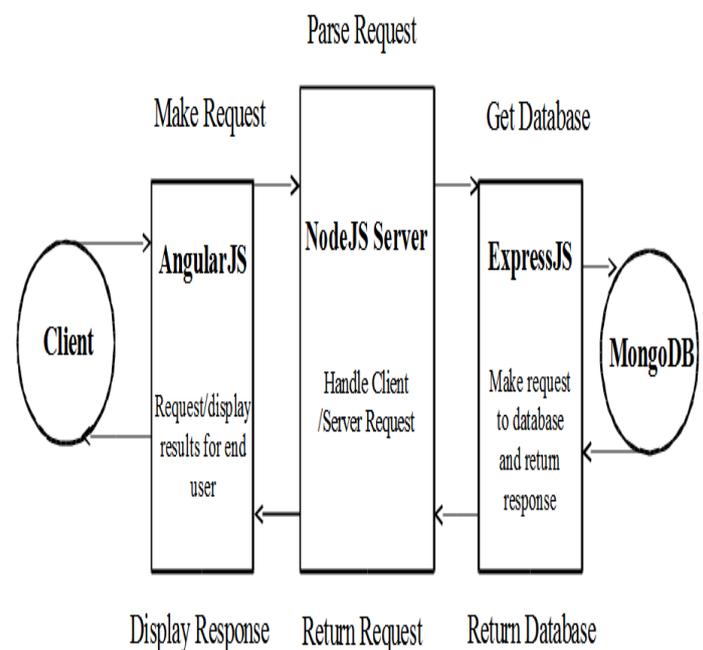
### 3.6 Node JS

NodeJS[5] is an open-source, cross-platform runtime environment which is used for developing server-side Web applications. Node.js is not a JavaScript framework implicitly, although its applications are written in JavaScript and are run within the NodeJS runtime on a variety of platforms.

Node.js provides an event-driven architecture and a non-blocking I/O API designed to optimize the throughput and scalability of the applications for real-time Web applications. It uses Google V8 JavaScript engine in order to execute code, and a huge percentage of the basic modules are written in JavaScript. It also contains a built-in library, which allows applications to work as a stand-alone Web server.

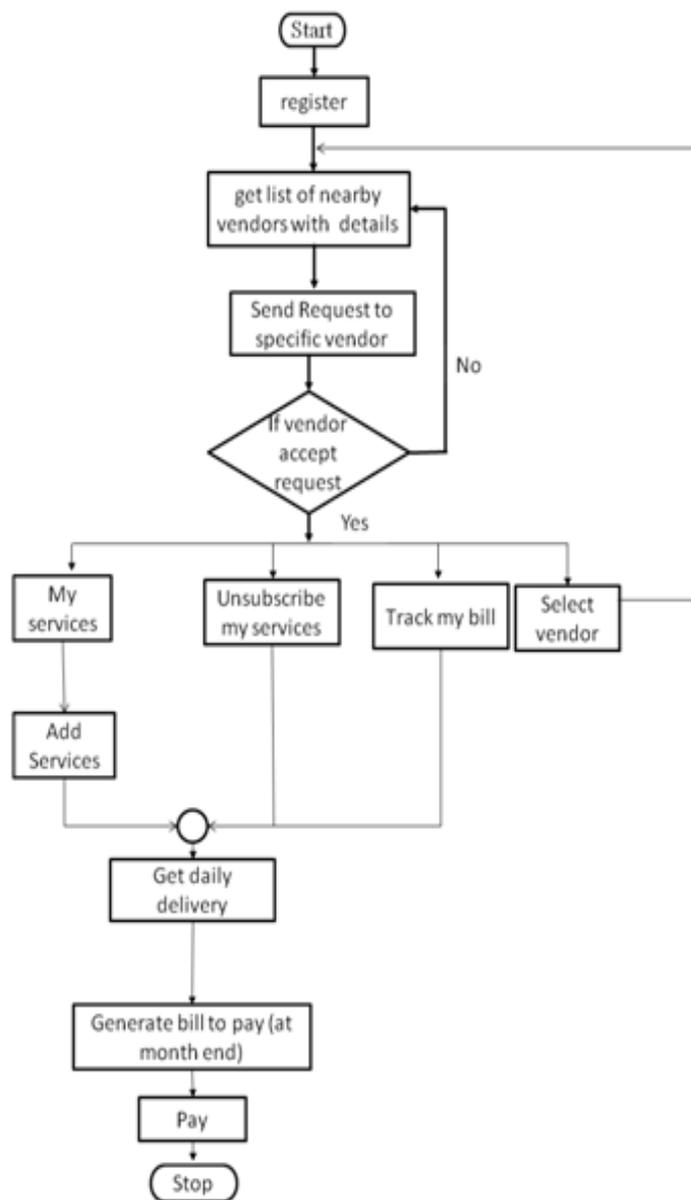
In the proposed system, Node JS is the platform which is used for doing the server side development.

The interaction between the front end and back end technologies are shown in Figure 1.



**Fig-1:** Architecture

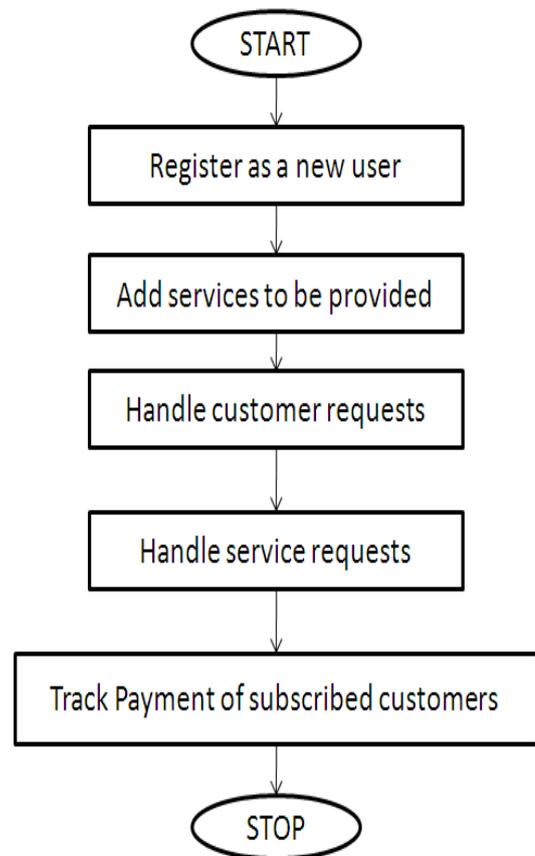
#### 4. FLOWCHART OF PROPOSED SYSTEM



**Fig-2:** Flowchart for customer

Customer will first register with this application. While registering he/she will need to fill his/her details like name, address, email id, contact number.

After filling this information, he/she will get the list of nearby vendors according to his/her address with their details. He/she will choose one of those vendors suitable for him/her and send request to that specific vendor. If vendor accepts his/her request, he/she will be redirected to his/her home screen where he/she will get various options as shown in the Figure 2.



**Fig-3:** Flowchart for vendor

The user when downloads the application first has to register as a vendor. According to Figure 3, the vendor should add services which he/she will provide to the customers. Next, he/she can accept customer requests and provide services to them. He/she can keep track of the bills of his/her customers for the products which were delivered to them on a monthly basis. Then he/she will take monthly payment from the customers for the services provided.

#### 5. ENTITY RELATIONSHIP DIAGRAM

The Entity Relationship diagram shown in Figure 4 gives the overview of the database of the proposed system in which there are two collections namely Customer collection and Vendor collection. Vendor provides customer the service which customer is subscribed to.

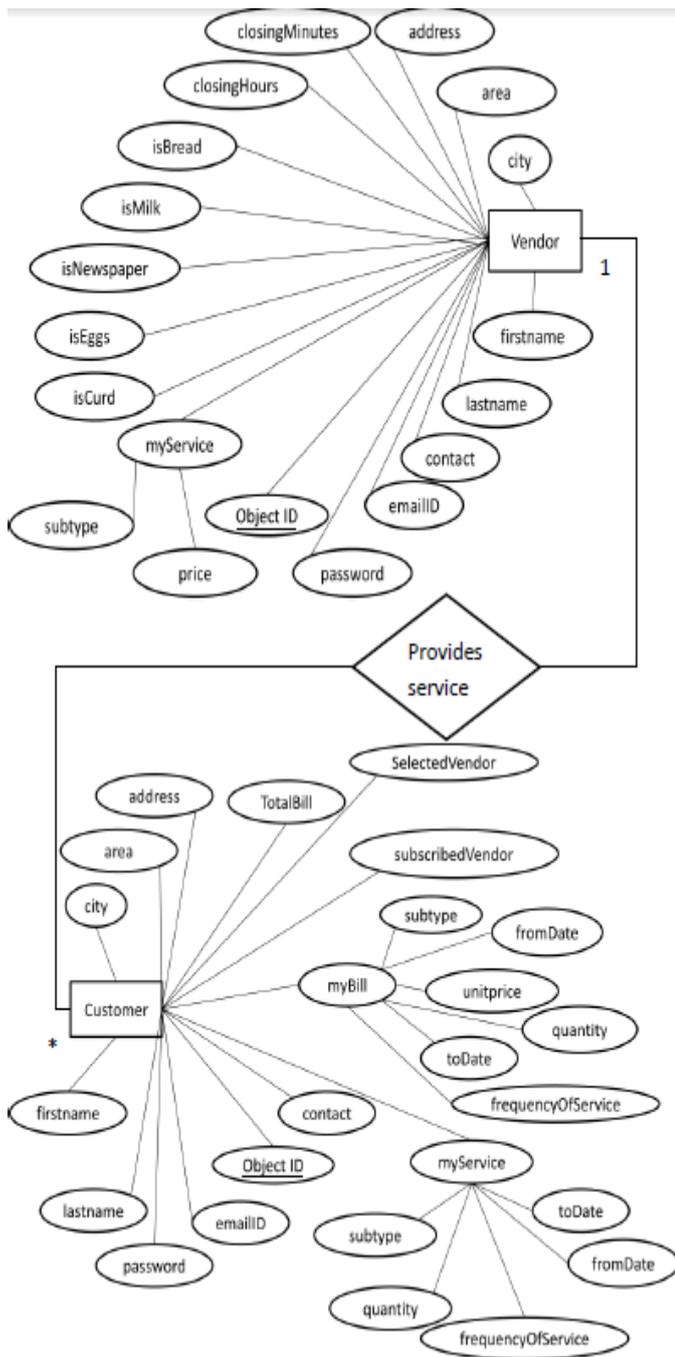


Fig-4: Entity Relationship Diagram

payment of the subscribed customers. This proposed application will also be contributing towards 'Make India Digital' campaign.

## REFERENCES

- [1] S. Hoberman, "Data Modeling for MongoDB", Publisher by Technics Publications, LLC 2 Lindsley Road Basking Ridge, NJ 07920, USA, ISBN 978-1-935504-70-2, 2014
- [2] Dan D. Gutierrez, Web database development for windows platforms, Prentice Hall, Upper Saddle River, 1999, pp. 48-50.
- [3] R. P. Padhy, M. R. Patra, S. C. Satapathy, "RDBMS to NoSQL: Reviewing Some Next-Generation Non-Relational Database's", International Journal of Advance Engineering Sciences and Technologies, Vol. 11, Issue No. 1, 015-030, 2011.
- [4] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format Interchange Format," 2014. [Online]. Available: <https://tools.ietf.org/html/rfc7159>
- [5] MEAN.io. (2015) MEAN — Full-Stack JavaScript Using MongoDB, Express, AngularJS, and Node.js. [Online]. Available: <http://mean.io/>

## 6. CONCLUSIONS

Thus this paper proposes to help the customers to save their time and efforts because the daily needs will be provided to them at their doorstep. It will also help them to manage their budget as they will be provided with the functionality to track their estimated bill. Also, this paper proposes to help the vendors by allowing them to manage their customers, their needs and also track the