RESEARCH ARTICLE OPEN ACCESS

Moped Service Provider Using Machine Learning

Swapnil Prachande, Tejas Nipankar, Dr. K.S. Wagh
Dept of computer engineering, Savitribai Phule University, Pune - India.

ABSTRACT
Now a days, technology is on a boost. People wish to live a luxurious life with minimum physical work. Here we provide a mobile application for 'Bike Service System'. This application is an android app which can be run on any android compatible tablets and mobile phones. The app will enable any bike user to search and communicate with any bike service center in the vicinity. The user can find the service center, get its location and check and select any of the services provided by the respective service center. The user can send request for pick and drop, appointment for servicing, test drive as well as accessories purchase to the dealer. The dealer processes these requests and gives a response back to the user through push messages. Thus, we are developing an application which goes hand in hand with the new age technology and characterizes user friendliness, in formativeness and time saving.
Keywords: **-** Android, Angular JS, MySQL, PHP, HTML, CSS3, Bootstrap

I. INTRODUCTION

Internet tends to be the backbone of all the technologies. The Bike Service Center Management System (BSCM) is a progressive step in the field of service centers. Any bike user can make use of such app to locate and communicate with the service centers in the vicinity. The proposed system can be used by any automobile user.

II. PROPOSED SYSTEM

The purpose of this project is to provide Bike or any other automobile servicing system more effectively than the existing system. There are some disadvantages of the existing service center management systems. These disadvantages are overcome by the bike service center management system. And it can be made handily available to every person. Previously people could not get help or locate the service centers conveniently in case of their car break-down or any other emergencies. Thus BSCM is proposed to assist people and fulfill their requirements easily

Advantage

- Easy to use because all Details of bike servicing will quickly available 24 x 7 on mobile.
- It can be easily accessed globally with help of Internet.
- Maintaining records will be easier because all details are stored in database and retrieved easily from it.
- Interactive and attractive design.
- Provides Alerts or Reminder by mobile app.
- Provides online booking of bike and servicing easily.
- Provides the user to pick and delivery the bike using mobile services
- User can easily pick the nearby services center

III. OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

Servicing Details:

It will provide all details of servicing of your bike including job card details and additional parts changing details. It provides details according to specific user. Online booking facility will be also available for customer convenience so one has not wait in queue.

Bike sales details:

It provides all specification & functionality of bikes. It also provides Sales details. It will provide facility for online bike booking. It will also provide delivery details.

System Features

In the life of the software development, problem analysis provides a base for design and development phase. The problem is analyzed so that sufficient matter is provided to design a new system. Large problems are sub-divided into smaller once to make them understandable and easy for finding solutions. Same in this project all the task are sub-divided and categorized.

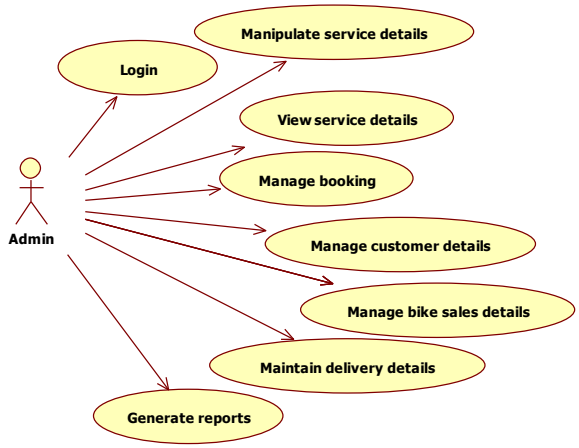

Fig 1: Admin Module of the application

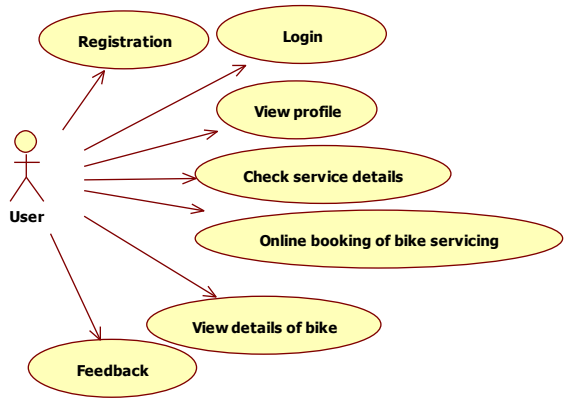

Fig 2: User Module of the application

IV. ARCHITECTURE DIAGRAM

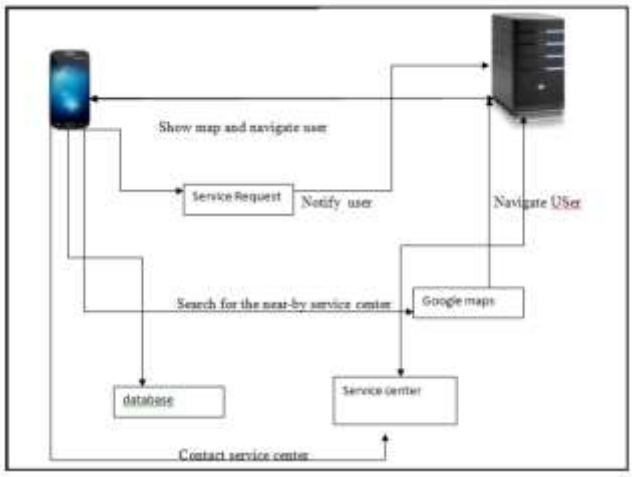

Fig 3: System Architecture

V. MOBILE APP DEVELOPMENT PROCESS

An effective app development process flow spans over six key phases. Regardless of the size and scope of your project, following this development process will make enterprise mobile app development initiative a success.

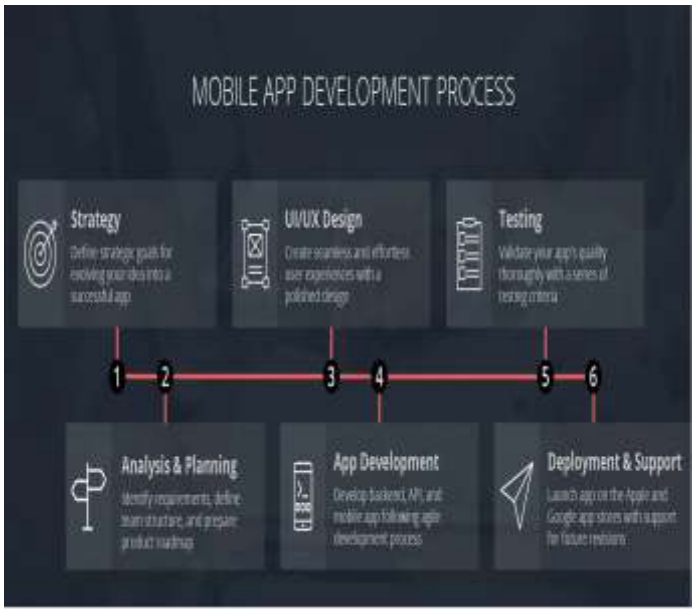

Fig 4: mobile-app-development-process-flow

1.Strategy

The first phase of the mobile app development process is defining the strategy for evolving your idea into a successful app. You may include a more significant part of this in your overall enterprise mobility strategy. As one app's objectives may differ from another, there is still an app-specific impact to the mobility strategy to address during the development process.

In this phase, you will:

- Identify the app users
- Research the competition
- Establish the app's goals and objectives
- Select a mobile platform for your app

2. Analysis and Planning

At this stage, your app idea starts taking shape and turns into an actual project. Analysis and planning begin with defining use cases and capturing detailed functional requirements.

After you have identified the requirements for your app, prepare a product roadmap. This includes prioritizing the mobile app requirements and grouping them into delivery milestones. If time, resources or costs are a concern, then define your minimum-viable-product (MVP) and prioritize this for the initial launch.

Part of the planning phase includes identifying the skills needed for your app development initiative. For example, iOS and Android mobile platforms use different development technology stacks. If your goals are to build a mobile app for both iOS and Android mobile platforms then, your mobile development team should include iOS developers and Android developers.

3. UI / UX Design

The purpose of an app's design is to deliver seamless and effortless user experiences with a polished look.

The success of a mobile app is determined based on how well users are adopting and benefiting from all its features. The goal for mobile app UI / UX design is creating excellent

user experiences making your app interactive, intuitive, and user-friendly. While polished UI designs will help with early adoption, your app must have intuitive user experiences to keep app users' engaged.

4. App Development

Planning remains an integral part of this phase in the mobile app development process. Before actual development/programming efforts start, you will have to:
define the technical architecture, pick a technology stack, and define the development milestones. A typical mobile app project is made up of three integral parts: back-end/server technology, API(s) and the mobile app front-end.

Back-End/Server Technology

This part includes database and server-side objects necessary for supporting functions of your mobile app. If you are using an existing back-end platform, then modifications may be needed for supporting the desired mobile functionality.

API

An Application Programming Interface (API) is a method of communication between the app and a back-end server/database.

Mobile App Front-End

The front-end is the native mobile app an end-user will use. In most cases, mobile apps consist of interactive user experiences that use an API and a back-end for managing data. In some cases, when an app needs to allow users to work without internet access, the app may utilize local data storage.
You can utilize almost any web programming language and databases for the back-end. For native mobile apps, you have to choose a technology stack required by each mobile OS platform. iOS apps can be developed using Objective-C or Swift programming language. Android apps are primarily built using Java or Kotlin.
There is more than one programming language and technology stack for building mobile apps —the key is picking a technology stack that is best suited for your mobile app.
Mobile technologies advance much faster with new versions of mobile platforms. Furthermore, new mobile devices are released every few months. With platforms and devices rapidly changing, agility is essential for building mobile apps within timelines and budgets. If time-to-market is a priority, use an agile development approach. This approach supports frequent software releases with completed functionality. Defining development milestones as part of the agile development plan supports developing your mobile application in iteration.
As each development milestone completes, it is passed on to the app testing team for validation.

5. Testing

Performing thorough quality assurance (QA) testing during the mobile app development process makes applications stable, usable, and secure. To ensure comprehensive QA testing of your app, you first need to prepare test cases that address all aspects of app testing.

Similar to how use cases drive the process of mobile app development, test cases drive mobile app testing. Test cases are for performing test steps, recording testing results for software quality evaluation, and tracking fixes for retesting. A best practice approach is involving your QA team in the Analysis and Design stages. The familiarity with your app's functional requirements and objectives will help produce accurate test cases.

Your app should undergo the following testing methods, to deliver a quality mobility solution.

- User Experience Testing:
  A critical step in mobile app testing is to ensure that the final implementation matches the user experience created by the app design team. Visuals, workflow, and interactivity of your app are what will give your end users first-hand impression of your app. Make sure that your app employs consistent fonts, style treatments, color scheme, padding between data, icon design, and navigation. Ensuring that your app matches the original design guidelines will have a direct impact on its user adoption

- Functional Testing:
  The accuracy of your mobile app functionality is critical to its success. It's difficult to predict every end user's behavior and usage scenario.

  The functionality of your app should be tested by as many users to cover as many potential testing conditions as possible. You might be surprised to catch bugs when two different users test the same feature but get varied outcomes. For example, both users can fill out the same form, but they both might enter different data—which could lead to discovering a defect.

  The purpose of functional testing is to ensure that users can use your app's features and functionality without any issues. It can be broken down further into system testing (the app working as a whole), and unit testing (individual functions of the app operating correctly).

  If you are building an app for iOS and Android mobile platforms, then your functional testing should include a feature comparison between both versions of your mobile app.

- Performance Testing:
  There are many quantitative criteria to use for measuring the performance of your app.

  - How well is your app responding to the user requests?
  - How fast are the app's screens loading?
  - Is your app draining the phone battery or causing memory leaks?
  - Does your app leverage network bandwidth efficiently?

- Is the size of your app bigger than what it should be?

Even when your app passes basic performance criteria, test the app, API, and backend for load by simulating the maximum number of concurrent users. Your app should be able to handle the load and perform well even when usage spikes.

- Security Testing:

Security is of utmost concern for enterprise mobile apps. Any potential vulnerability can lead to a hack. Many companies hire outside agencies to perform thorough security testing on their applications. Your QA and development teams can take a few simple measures to make your app secured.

If your app requires users to log in, these log in sessions should be tracked on the device and the backend. User sessions should be terminated by the system when a user has remained idle for an extended time (typically ten mins or less on a mobile app). If your app stores user credentials on the device to make it convenient for them to re-login, then you must ensure using a trusted service. For example, the development platform for iOS apps provide the Keychain feature that can be used for storing a user's account details for a specific app.

Data entry forms within your mobile app should be tested to ensure there is no data leakage.

- Device and Platform Testing:

On average, new mobile devices enter the market every 12 months with new hardware, firmware, and design. Mobile operating systems are updated every few months.

Multiple mobile device manufacturers like Samsung, LG, HTC, Motorola use the Android platform, but they customize the platform for their mobile devices (since Android is open source). The devices come in different sizes and shapes

Compare that to Apple, which has a lot more controlled environment, since they control both hardware and the OS. However, there are multiple iPhone & iPad (Apple iOS) devices out on the market.

This is where testing during the mobile app development process differs significantly from web app testing. You can get away by testing your web app just on the Chrome browser in a Windows environment. But your mobile app has to be tested on multiple mobile devices or device simulators to ensure smooth working of your app for all users.

The complexity of mobile app testing on all mobile devices, ongoing support costs, and headaches of mobile device management are primary reasons why companies tend to build their enterprise mobile apps for a single mobile platform (and often provide mobile devices to their users). In our experience, most companies tend to develop their enterprise mobile app first with Apple's iOS mobile platform; only where needed they build an app for the Android platform.

Testing is imperative to an app's future success; it encompasses a substantial section of our overall mobile app development process. Having a comprehensive mobile testing strategy is a must for delivering a quality mobile app.

During the testing phase, there are many ways for distributing your app development builds to the testers. The most common approach with iOS apps is using the Testflight and for Android apps via email or Over The Air (OTA) installs.

6. Deployment & Support

Releasing a native mobile app requires submitting your app to the app stores, Apple App Store for iOS apps and Google Play for Android apps. However, you will need a developer account with Apple App Store and Google Play Store before launching your mobile app.

An app's release in the app store requires preparing metadata including:

- Your app's title
- Description
- Category
- Keywords
- Launch icon
- App store screenshots

Once submitted in the Apple App Store, iOS apps go through a review process which may take from a few days to several weeks depending on the quality of your app and how closely it follows Apple's iOS development guidelines. If your app requires users to log in, then you will need to provide Apple with a test user account as part of the release process.

There isn't any review process with Android apps, and they become available in the app store within a few hours of submission.

VI. K MEANS CLUSTERING – INTRODUCTION

k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

The problem is computationally difficult (NP-hard); however, efficient heuristic algorithms converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of

Gaussian distributions via an iterative refinement approach employed by both k-means and Gaussian mixture modeling. They both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the Gaussian mixture model allows clusters to have different shapes.

The unsupervised k-means algorithm has a loose relationship to the k-nearest neighbor classifier, a popular supervised machine learning technique for classification that is often confused with k-means due to the name. Applying the 1-nearest neighbor classifier to the cluster centers obtained by k-means classifies new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.

History:

The term "k-means" was first used by James MacQueen in 1967, though the idea goes back to Hugo Steinhaus in 1956. The standard algorithm was first proposed by Stuart Lloyd of Bell Labs in 1957 as a technique for pulse-code modulation, although it was not published as a journal article until 1982.In 1965, Edward W. Forgy published essentially the same method, which is why it is sometimes referred to as the Lloyd–Forgy algorithm.

Discussion:

Three key features of k-means that make it efficient are often regarded as its biggest drawbacks:

- Euclidean distance is used as a metric and variance is used as a measure of cluster scatter.
- The number of clusters k is an input parameter: an inappropriate choice of k may yield poor results. That is why, when performing k-means, it is important to run diagnostic checks for determining the number of clusters in the data set.
- Convergence to a local minimum may produce counterintuitive ("wrong") results.

Applications:

k-means clustering is rather easy to apply to even large data sets, particularly when using heuristics such as Lloyd's algorithm. It has been successfully used in market segmentation, computer vision, and astronomy among many other domains. It often is used as a preprocessing step for other algorithms, for example to find a starting configuration.

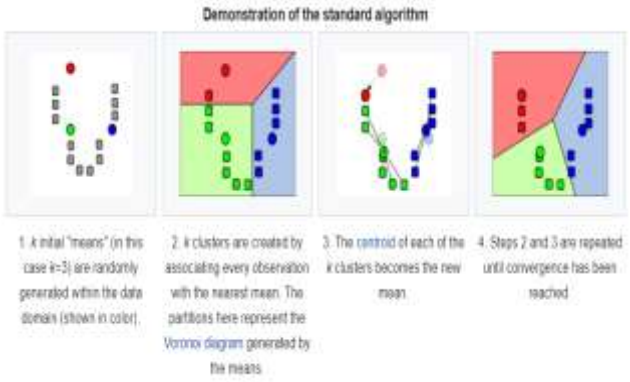

Fig 5: output screenshot

VII. RESULTS

Fig 6: output screenshot

Fig 7: output screenshot

CONCLUSION

The proposed paper shows the flow, structure and working of the Bike Service Center Management (BSCM) system. BSCM is user friendly i.e. easy to use. It is free of cost on android store. Thus, it is time a time saving as well as cost efficient application. So, we can conclude that the proposed system can be used to reduce human efforts and luxuriate human lives, hand in hand, with the modern technology.

REFERENCES

i. Adler M. J. (2009). Harvard Business Review. Health Care Requires Big Changes to Complement New IT, 87, 20.

ii. Appolinário, F. (2004). Dicionário de MetodologiaCientífica: um Guiapara a Produção do ConhecimentoCientífico.

iii. Arnhold, l.; Schmidt, S. &Bohnenberger, M.C. (2008). Seleção e Implantação de ERP emumaCooperativaMédica no RS: Um Estudo de Caso. In: EncontroAnual da ANPAD. 32, Rio de Janeiro, 2008. Anais... Rio de Janeiro: ANPAD.

iv. Bahensky, J. A.; Jaana, M. & Ward, M. M. (2008). Health Care Information Technology in Rural America: Electronic Medical Record Adoption Status in Meeting the National Agenda. The Journal of Rural Health.

v. Balfour III, D.C. (2009). et al. Health Information Technology — Results From a Roundtable Discussion. JMCP Supplement to Journal of Managed Care Pharmacy. (January/February).