

LOCAL SERVICE SEARCH ENGINE MANAGEMENT SYSTEM

**PURITY AKINYI ODHIAMBO**



**2024**

**INFORMATION TECHNOLOGY SYSTEMS PROPOSAL SUBMITTED TO THE DEPARTMENT OF BUSINESS ADMINISTRATION FOR THE AWARD OF BACHELORS DEGREE OF BUSINESS INFORMATION TECHNOLOGY OF JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY**

# **DECLARATION**

Table of Contents

[**CHAPTER ONE** 3](#_Toc162220730)

[**INTRODUCTION** 3](#_Toc162220731)

[**1.1** **Background of the System** 3](#_Toc162220732)

[**1.2** **Statement of the Problem** 5](#_Toc162220733)

[**1.2.1** **Proposed Solution** 6](#_Toc162220734)

[**1.3** **System Objectives** 7](#_Toc162220735)

[**1.3.1** **General Objective** 7](#_Toc162220736)

[**1.3.2** **Specific Objectives** 7](#_Toc162220737)

[**1.4** **System Functionalities** 7](#_Toc162220738)

[**1.5** **System Justification** 8](#_Toc162220739)

[**1.5.1** **Service Providers** 8](#_Toc162220740)

[**1.5.2** **Service Seekers** 8](#_Toc162220741)

[**1.6** **Scope of the System** 9](#_Toc162220742)

[**CHAPTER TWO** 10](#_Toc162220743)

[**LITERATURE REVIEW** 10](#_Toc162220744)

[**2.1** **Introduction** 10](#_Toc162220745)

[**2.2** **Empirical Review** 10](#_Toc162220746)

[**2.2.1** **Existing System Architectures** 10](#_Toc162220747)

[**2.2.2** **Proposed System Architecture** 14](#_Toc162220748)

[**2.3** **System Development Requirements** 18](#_Toc162220749)

[**2.3.1** **Software Requirements** 18](#_Toc162220750)

[**2.3.2** **Hardware Requirements** 18](#_Toc162220751)

[**CHAPTER THREE** 20](#_Toc162220752)

[**SYSTEM DEVELOPMENT METHODOLOGY** 20](#_Toc162220753)

[**3.1** **Introduction** 20](#_Toc162220754)

[**3.2** **System Development Model** 20](#_Toc162220755)

[**3.3** **Fact Finding Techniques** 25](#_Toc162220756)

[**3.3.1 Surveys and Questionnaires** 25](#_Toc162220757)

[**3.3.2 Interviews** 26](#_Toc162220758)

[**3.4** **Feasibility Study** 28](#_Toc162220759)

[**3.4.1** **Technical Feasibility** 28](#_Toc162220760)

[**3.4.2** **Operational Feasibility** 28](#_Toc162220761)

[**3.4.3** **Legal and Regulatory Feasibility** 29](#_Toc162220762)

[**3.4.4** **Social Feasibility** 30](#_Toc162220763)

# **CHAPTER ONE**

# **INTRODUCTION**

## **Background of the System**

Marketplaces are entities that enable transactions between buyers and sellers; importantly, they operate as a third party to the principal transaction (Vidal, 2019). As a result, the marketplace is neither a seller nor a buyer, and hence does not own the merchandise or services traded on the platform. If they did, the entities would be more like traditional merchants than marketplaces. As a result, it would be erroneous to claim that well-known digital markets such as Uber, Etsy, and AirBnB provide transportation, handmade items, or lodging. Rather, they encourage interactions between drivers and riders, artists and buyers, and hosts and guests (Marketplaces, 2014).

Customers can use online service marketplaces to identify service providers in their desired regions for a wide range of jobs such as carpentry, plumbing, electricity, cleaning, gardening, and so on. Some of these markets include TaskRabbit, Thumbtack, and Handy, which strive to link clients with experts in a range of industries.

So, how do markets support such transactions, and what do they truly sell? Marketplaces sell lower transaction costs. According to Stanford professor and markets specialist Ramesh Johari, these transaction costs can be classified into the following types (Obrenović, 2021):

1. Costs of Search and Information: Search refers to the cost of connecting buyers and sellers, while information refers to the cost of gathering necessary information to make an informed choice on whether to proceed with a transaction.
2. Negotiation and Bargaining Costs: Bargaining and negotiation costs are the expenses incurred in determining what is being sold and at what price.
3. Costs of Policing and Enforcement: Policing and enforcement costs are the costs of ensuring that transactions are handled fairly and favorably for both parties.

A marketplace need not mitigate costs from all three dimensions, but we cannot imagine a marketplace that does not mitigate costs along at least one. In ancient Greece, the agora marketplace was the focal point of athletic, artistic, spiritual, and political life, and its geographical and cultural importance allowed it to cut transaction costs. New kinds of media have enabled virtual marketplaces, which connect buyers and sellers regardless of their physical location. Newspaper classified advertising were the first virtual marketplaces, accounting for around 40% of the ad industry's revenue in 2000 (Marketplaces, 2014).

The democratization of the Internet in the 1990s paved the way for a new sort of commerce: e-Commerce. If platforms selling retail items were the first to embrace e-commerce, platforms delivering services followed a decade later. This is the example of the giant Airbnb, which reports 4.7 billion in income in 2019 after little over ten years of operation. Other unknowns have become household brands because to the exceptional growth prospects that the marketplace model provides for services.

The word "service marketplace" implies that this type of platform would exclusively allow the sale of services. But it isn't. There are three sorts of service marketplaces: rental marketplaces, like Airbnb; network marketplaces, which connect several parties and enable data exchange; and marketplaces that offer services (Cocolabs, 2021).

Local Service Search Engine Management Systems (LSSEMS) are web-based applications which provide technology-based platforms to users to take care of their daily needs. They can help users get the serviceman of locality at doorstep. They act as a platform for users and service providers to interact with each other about delivering the desired service. The main purpose of LSSEMS is to solve the problem of users who search for servicemen in their own locality by providing a platform for users and service people such as maids, tutors, plumbers etc. (YogeshIJTSRD, 2021).

## **Statement of the Problem**

The demand for community-based services has increased in today’s urban environment, covering a wide range of areas such as home care, personal care, education, etc. but service seekers and providers face huge challenges in obtaining and understanding these services (Smith, 2019). The existing landscape lacks a unified way to find local businesses, so consumers rely on multiple sources, word of mouth, or general online research (Johnson, 2020). This expansion provides it is difficult to identify and effectively implement appropriate services in their local communities (Davis, 2018 ).

Also, users often find it difficult to determine the reliability and trustworthiness of local service providers. Service seekers struggle to make educated decisions because there is no standardized method of verification or a standardized way to screen and compare users.

Service providers, on the other hand, find it difficult to connect with their target audience and stay online. The absence of a specific platform to demonstrate capacity, manage constituencies and perform tasks leads to inefficiencies (Brown, 2020).

Many local entrepreneurs deal with geographical obscurity, making it difficult for them to connect with potential customers in the surrounding neighborhoods. As a result, service providers lose opportunities, while service seekers are left with limited local options.

The lack of a standardized and secure payment mechanism in the local service discovery system creates problems for service seekers and suppliers (Jones, 2020). These differences often lead to problems, delayed projects, and potential security risks.

### **Proposed Solution**

The Local Service Search Engine Management System will provide service seekers with a centralized location to search for local applications. Users can easily find and search for relevant services in their area instantly thanks to the user interface and comprehensive search capabilities

To address trust and reliability concerns, the system will establish a robust verification process for service providers. This can include background checks, credentials verification, and a customized profiling process. Additionally, a comprehensive user review and comparability system will be implemented, enabling service seekers to make informed decisions based on the experience of others.

The system will provide service providers with a personal profile where they can draw on their capabilities, expertise and portfolio. The software allows providers to easily manage their schedules, appointments and stakeholder interactions. Financial transactions will be easier thanks to integrated billing and secure payment mechanisms.

The system will use geolocation services that will allow local service providers to locate in their target geographic area. Users can search and prioritize service offerings based on their location, creating greater connections between local businesses and the community.

The local service search engine management system will have secure and standard payment gateways. This will ensure smooth, transparent and secure financial transactions between service providers and suppliers. To make things easier for users, the system can provide an e-wallet so that payments can be done automatically and securely.

## **System Objectives**

### **General Objective**

To develop a local service search engine management system for local service people and service seekers around Nairobi.

### **Specific Objectives**

1. To determine a safe and secure way for transactions between local service people and service seekers
2. To determine how service people and service seekers within the same geographical area will easily find each other
3. To determine how trustworthiness and credibility will be upheld by both the service people and service seekers

## **System Functionalities**

1. An e-wallet will be a functionality applied to facilitate a safe and secure method of funds transfer between the service person and the service seeker for the services provided.
2. A search and filter feature shall be applied in order to filter the service providers and seekers according to their location therefore narrowing the number of users to only those that are relevant to both the user and service provider
3. A thorough vetting of service providers will be done in order to ensure their credibility and their credentials taken. In addition to that, a ratings and reviews feature shall be added not only for the service providers, but also for the service seekers in order to endure the safety of both parties. This shall help both parties make informed decisions as they surf the system.

## **System Justification**

### **Service Providers**

The system will offer a dedicated platform where local service providers can showcase their skills, expertise and services provided, which will increase their visibility thus increasing their chances of being discovered by relevant service seekers. The system will also help them manage their business through appointment scheduling and management. They will be able to communicate directly with clients and update their availability seamlessly through the platform. The ratings and reviews feature within the system will help service providers build a good reputation for themselves within the platform thus making their chances of discovery by clients to be higher and thus more jobs for them. The e-wallet feature, through which transactions will be made, will be of help to the service provider whereby their transactions will be automated as soon as the service is marked done by both parties. This will increase security in terms of payment forfeiture by the clients. Through the system, service providers will be able to market and promote their services easily and to a vast number of service seekers within their geographical locality, thus increasing their chances of being discovered by potentially relevant clients.

### **Service Seekers**

The system will be of much use to the service seekers whereby they will be able to find a vast group of verified and credible service providers, narrowed down to those within their geographical locality, within one platform. This will not only make it easier for them to make a selection, but will be so much easier for them to meet without need for much travel. The ratings and reviews feature will allow service seekers to make informed decisions while selecting their preferred service provider, therefore building trust. They will also be able to write their own reviews regarding the services that they have received. The system will allow for convenient booking and scheduling of services with the service providers. In addition to that, the service seeker will be able to communicate directly with the service provider where they can discuss and negotiate prices. They will also receive updates, and discuss specific service requirements, which will in turn increase the overall customer satisfaction.

## **Scope of the System**

The Local Service Search Engine Management System is a platform designed to connect local service providers to relevant service seekers within their geographical location about delivering the desired service. Service providers will be able to use the platform to create their profiles and showcase their skills, expertise and services offered in order to attract the attention of a vast number of service seekers. The service seeker shall be able to use the system to find and select vetted and relevant service providers within their locality. They will be able to negotiate prices, rate and review services provided and pay for the services received through the secure platform.

# **CHAPTER TWO**

# **LITERATURE REVIEW**

## **Introduction**

This section aims to establish the theoretical framework and contextual background necessary for understanding the design, functionality, and effectiveness of local service search engines. By reviewing relevant literature, researchers and developers can identify gaps in current knowledge, explore best practices, and leverage insights to enhance the performance and user experience of their systems.

## **Empirical Review**

### **Existing System Architectures**

#### **TaskRabbit**

TaskRabbit operates on a client-server system architecture. In this architecture, the system is divided into two main components: the client, which is the user interface through which users interact with the platform, and the server, which processes requests from clients and provides the necessary resources and services. The client-side of TaskRabbit’s system architecture consists of the web or mobile application that users access to post tasks, browse available taskers, and communicate with them. The client application sends requests to the server for tasks to be performed and receives responses with relevant information such as tasker profiles, availability, and pricing. On the server-side, TaskRabbit’s system architecture includes various components such as databases to store user information and task data, application servers to process user requests and business logic, and communication protocols to facilitate interactions between clients and servers. The server also handles authentication and authorization processes to ensure secure access to user accounts and sensitive information.

The user interface of TaskRabbit is designed to be intuitive and user-friendly, allowing users to easily navigate the platform and find the services they need. Upon visiting the TaskRabbit website or mobile app, users are greeted with a clean and organized layout that presents them with options to either post a task or browse available taskers. The platform provides filters and search functions that enable users to quickly narrow down their search based on criteria such as location, price range, and task category. Task posters can create detailed task listings that outline the requirements, budget, and timeline for the task. They can also view profiles of potential taskers, read reviews from previous clients, and communicate with taskers through the messaging system. Taskers, on the other hand, can browse available tasks in their area, submit offers to complete tasks, and communicate with task posters to clarify details.

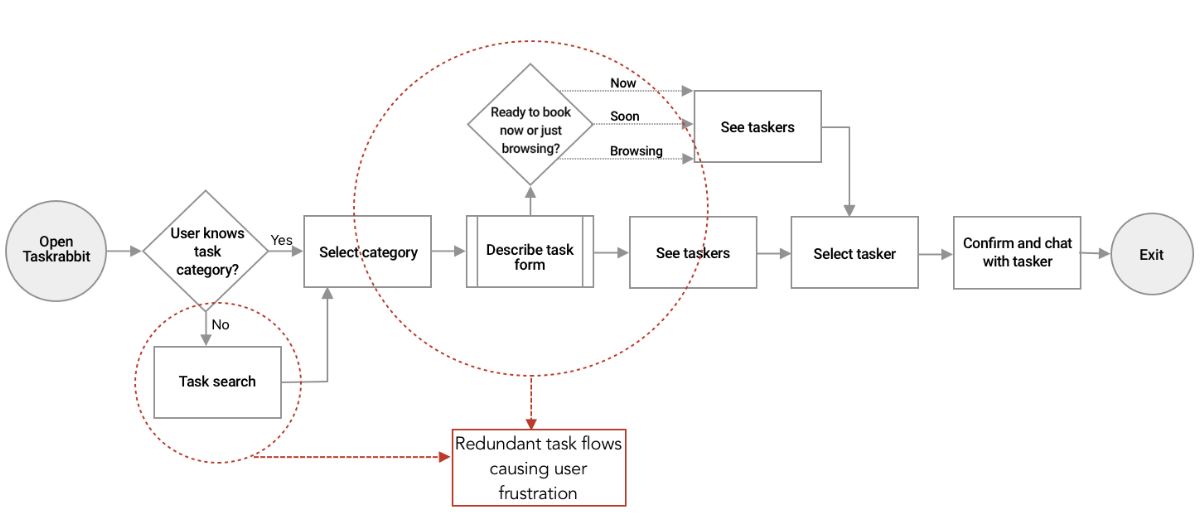


Figure 1 TaskRabbit task flow

#### **Handy**

The system architecture used by Handy is a two-sided marketplace platform. This type of architecture connects service providers (such as cleaners, handymen, and other professionals) with customers who require their services. The platform acts as an intermediary, facilitating transactions between the two parties and providing tools for scheduling, payment processing, and communication. Handy’s system architecture typically includes a front-end interface for customers to browse services, book appointments, and make payments. On the other side, service providers have access to a separate interface where they can manage their schedules, accept job requests, and communicate with customers. Behind the scenes, there is a central database that stores information about users, services, transactions, and reviews. The platform also incorporates algorithms for matching service providers with customer requests based on factors such as location, availability, and ratings. Additionally, Handy’s system architecture may include features for handling customer support inquiries, managing disputes between users, and collecting feedback to improve the overall user experience.

The app features a clean and simple layout with clear categories for different services, making it easy for users to find what they are looking for. Users can quickly book appointments, view service provider profiles, and manage their bookings all within the app.One of the key features of the Handy user interface is its rating and review system. Customers can rate their service providers and leave reviews, which helps other users make informed decisions when choosing a provider. This transparency and feedback mechanism contribute to building trust among users and maintaining service quality standards.

Another important aspect of the Handy user interface is its payment system. Users can securely pay for services within the app using various payment methods, adding convenience to the overall user experience. The app also allows users to track their service provider’s location in real-time, providing them with updates on their arrival time.

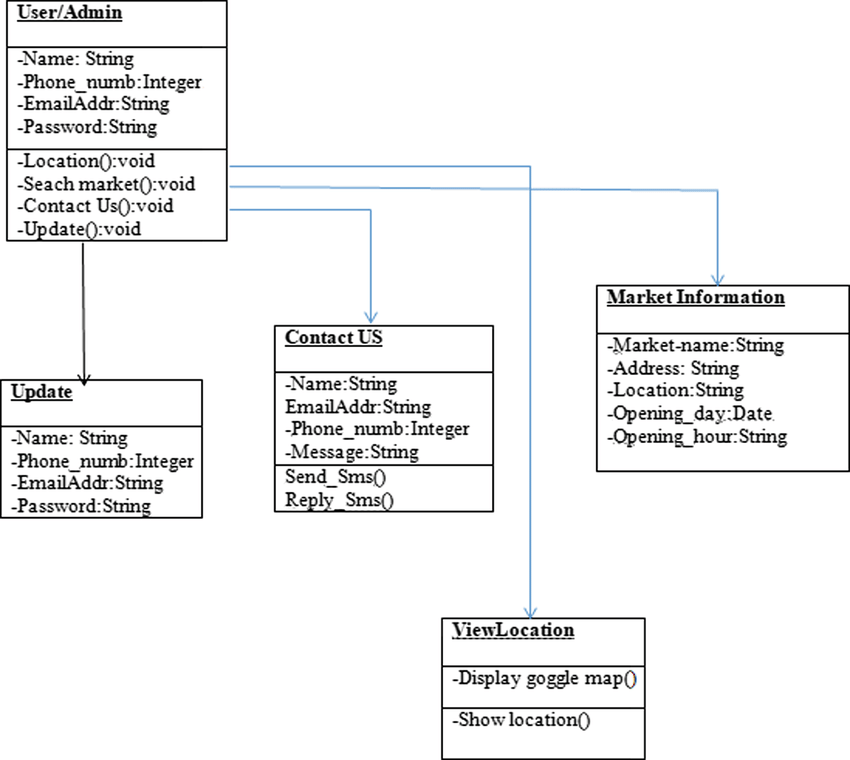


Figure 2 Handy Class Diagram

### **Proposed System Architecture**

The proposed architecture comprises several key components, each serving specific functions within the system as shown below:

#### **User Interface Layer**

The user interface layer is the front-end component of the system responsible for interacting with users. It will include:

##### **Web Interface:**

A responsive web application accessible via browsers on desktop and mobile devices.

##### **User Dashboard:**

A personalized dashboard for registered users, facilitating task management, messaging, and other interactions.

#### **Application Layer:**

The application layer contains the core logic of the system, including various modules and services:

##### **Search Engine:**

Will utilize advanced search algorithms to match service seekers with relevant service providers based on location, service type, ratings, and other criteria.

##### **User Authentication:**

Will manage user registration, login, and authentication processes, ensuring secure access to the system's features.

##### **Task Management:**

Will facilitate the creation, assignment, tracking, and completion of tasks between service seekers and providers.

##### **Messaging System:**

Will enable real-time communication between users, allowing them to discuss task details, negotiate terms, and provide updates.

##### **Payment Gateway Integration:**

Will integrate with third-party payment gateways to facilitate secure and seamless transactions between users specifically an e-wallet.

##### **Rating and Review System:**

Will allow users to rate and review service providers based on their experiences, contributing to the platform's trustworthiness and reputation.

#### **Data Layer**

The data layer stores and manages the system's data, including user profiles, task details, messages, reviews, and system configurations. It comprises:

##### **Relational Database:**

Will store structured data in tables, ensuring efficient retrieval and management of user and task-related information.

##### **NoSQL Database:**

Will store unstructured data such as user-generated content (e.g., reviews, messages) and provides scalability and flexibility for handling large volumes of data.

##### **File Storage:**

Will store multimedia files (e.g., images, documents) associated with tasks and user profiles, allowing for seamless content delivery and management.

#### **Integration Layer**

The integration layer facilitates communication and data exchange between different system components and external services:

##### **API Gateway:**

Will provide a unified interface for internal services and external integrations, enabling seamless communication and data exchange.

##### **Third-Party Integrations:**

Will integrate with external services such as mapping APIs for location-based services, payment gateways for transaction processing, and messaging APIs for real-time communication.

The proposed system architecture provides a robust foundation for developing and deploying the Local Service Search Engine Management System. By leveraging modern technologies and best practices, the architecture aims to deliver a reliable, scalable, and user-friendly platform for connecting service seekers with local service providers.

A diagram illustrating the above proposed architecture is shown below:

Web Interface

User Dashboard

API Gateway

Third-Party Integrations

Relational Database

NoSQL Database

File Storage

Search Engine

User Authentication

Task Management

Messaging System

Payment Gateway

Ratings and Review System

**User Interface Layer**

**Application Layer**

**Data Layer**

**Integration Layer**

Figure 3 Architecture of the Proposed System

## **System Development Requirements**

### **Software Requirements**

#### **Integrated Development Environment (IDE):**

Visual Studio Code Will be the most suitable IDE for the design of the System

#### **Version Control System:**

Git shall be used for managing source code repositories, versioning, and collaboration.

#### **Database Management System (DBMS):**

MySQL will fit the data storage requirements and preferences.

#### **Programming Languages and Frameworks:**

##### **Backend:**

The server-side logic and APIs that will be suitable for the design of the system will be PHP

##### **Frontend:**

For building the user interface, HTML, CSS, and JavaScript along with frontend frameworks/libraries like React.js, Angular, or Vue.js.

### **Hardware Requirements**

#### **Development Workstations:**

High-performance desktop or laptop computers for developers to write, build, and test code

#### **Server Infrastructure:**

##### **Web Servers:**

Depending on the expected traffic and load, multiple web servers may be needed to handle incoming requests. These could be physical servers or virtual machines.

##### **Database Servers:**

Separate servers or instances for hosting the database management system (DBMS).

##### **Load Balancers:**

Hardware load balancers or software load balancers to distribute incoming traffic across multiple web servers for load balancing and high availability.

#### **Storage:**

Storage Area Network (SAN) and Network Attached Storage (NAS) for storing application data, user files, and backups.

Solid State Drives (SSDs) and Hard Disk Drives (HDDs) depending on performance and storage requirements.

# **CHAPTER THREE**

# **SYSTEM DEVELOPMENT METHODOLOGY**

## **Introduction**

This section outlines the structured approach I will follow to design, build, and deploy the platform. Through systematic planning, development, testing, and iteration, this methodology ensures the efficient and effective delivery of a robust and user-friendly solution. By adhering to best practices and industry standards, aim to create a scalable and reliable system that meets the needs of both service seekers and providers.

## **System Development Model**

The Agile development model will be effectively utilized in the development of the Local Service Search Engine Management System to promote flexibility, collaboration, and iterative improvement throughout the development process. Below, I'll detail how Agile principles will be applied at each stage of development:

1. **Initiation:**

Planning Meeting: The development team, stakeholders, and project manager hold a planning meeting to define project goals, scope, and requirements. They identify key features and prioritize them based on customer needs and business value.

1. **Planning:**

User Stories: Requirements are captured as user stories, each describing a specific functionality from the perspective of an end-user. These stories are written collaboratively by the development team and stakeholders.

Sprint Planning: The project backlog is broken down into smaller tasks, and sprint planning sessions are held to select the tasks for the upcoming sprint. The team estimates the effort required for each task and commits to completing them within the sprint duration.

1. **Execution:**

Sprint Execution: Development work begins with a sprint, typically lasting 1-4 weeks. The team works on the selected tasks, implementing features, and functionality according to the user stories.

Daily Standup Meetings: The team holds daily standup meetings to discuss progress, share updates, and identify any impediments or blockers. This promotes transparency, communication, and collaboration within the team.

1. **Review and Feedback:**

Sprint Review: At the end of each sprint, a sprint review meeting is conducted to demonstrate completed work to stakeholders. Feedback is gathered, and any necessary adjustments are made to the product backlog based on stakeholder input.

Retrospective: The team holds a retrospective meeting to reflect on the sprint, identify what went well and what could be improved, and make adjustments for future sprints. This continuous improvement process helps the team learn and adapt over time.

1. **Deployment:**

Incremental Deployment: Agile promotes the delivery of working software at the end of each sprint. Incremental updates and improvements are deployed regularly, allowing stakeholders to see tangible progress and provide feedback early in the development process.

An illustration of the above is shown below:

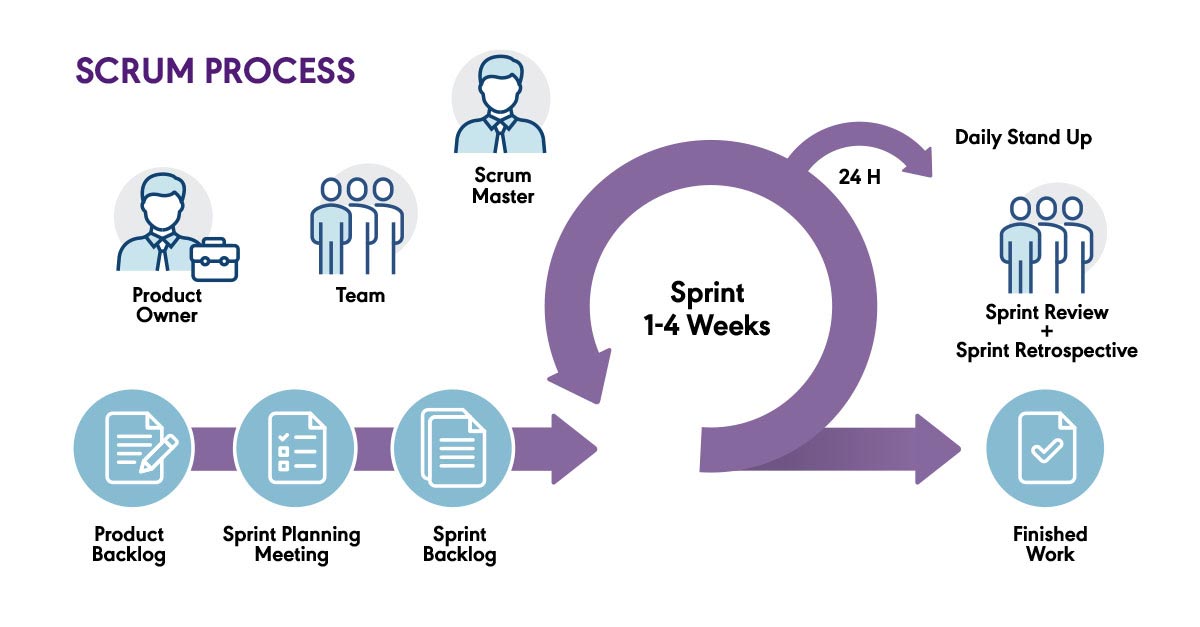


Figure 4 Agile Illustration

When you use an Agile methodology, especially the Scrum process, to develop a Local Service Search Engine Management System, there are some of the following benefits:

1. ***Flexibility and adaptation:***

Agile allows for flexible and iterative development, enabling the team to respond quickly to changes in requirements, market conditions, or user feedback. This is particularly beneficial in the rapidly evolving local service industry, where customer needs and market trends can change frequently.

1. ***Focus on Customer Satisfaction:***

Agile prioritizes customer satisfaction by delivering features and functionality that provide the most value to users. By continuously assessing and reprioritizing the project backlog based on customer feedback and changing market conditions, the team can ensure that the system meets the evolving needs of service seekers and providers.

1. ***Improved Quality and Transparency:***

Agile encourages continuous testing, integration, and validation throughout the development process. By breaking down work into smaller, manageable tasks and conducting regular code reviews and testing, the team can identify and address issues early, leading to higher-quality software.

1. ***Risk Mitigation:***

Agile reduces project risk by breaking down work into smaller iterations and delivering incremental value. This allows the team to identify and address potential issues early, minimizing the impact of unforeseen challenges or changes in project scope.

1. ***Continuous Improvement:***

Agile methodologies foster a culture of continuous improvement. The iterative nature of Agile allows for frequent retrospectives, where the development team reflects on previous runs, identifies areas for improvement, and implements necessary changes This allows for a more streamlined process and optimized over time, addressing any challenges, technical costs, or implementation issues

While the Agile methodology, especially the Scrum process, offers many advantages, it is important to consider potential drawbacks when using it to create an online bus ticketing system, here are some potential drawbacks:

1. ***Lack of Predictability:***

Agile methodologies prioritize adaptability and responsiveness to change, which can make it challenging to predict project timelines and deliverables. This can be a concern for stakeholders who require a clear timeline or have fixed deadlines. The iterative nature of Agile may lead to uncertainty regarding when specific features or functionalities will be implemented.

1. ***Scope creep:***

Agile methodologies accommodate changing requirements, but can lead to scope creep if not handled properly. Ongoing feedback and changes may result in project extensions that may affect project timelines and budgets. It is important to set clear boundaries and manage requirements effectively to avoid creeping into the impact of the development process.

1. ***Over-reliance on users:***

Agile methods rely heavily on the cooperation and feedback of the employees. In the case of online bus ticketing systems, it can be difficult for bus operators or passengers to get timely and consistent feedback from users. Absence or lack of user engagement can hinder the development process and hinder the ability to deliver a user-centric product.

1. ***Documentation and communication challenges:***

Agile methodologies prioritize functional software over complete documentation. While this can be useful in flexibility, it can also make it difficult for the system to manage detailed documentation. Additionally, the constant demand for communication and collaboration in Agile can be difficult when team members are distributed across geographic areas or at different times.

1. ***Learning style and team change:***

Adopting agile methods, especially if the development team is unfamiliar with them, may require a learning curve. For example, scrum practices can take time to fully understand and effectively implement. It is important to ensure that the team is properly trained and supported in the transition to Agile to maximize the benefits.

Many of these deficiencies can be mitigated through proper planning, effective project management and strong teamwork.

## **Fact Finding Techniques**

### **3.3.1 Surveys and Questionnaires**

The project team will distribute electronic surveys and questionnaires, both online and physically, to a wide number of users in JKUAT and around Juja and Nairobi CBD, in order to get uniform feedback on their needs, preferences, and viewpoints.

**Advantages**

1. ***Wide Reach***: By sending surveys and questionnaires to a lot of different parties, it was possible to simultaneously collect a variety of viewpoints.
2. ***Standardized Responses***: Standardized questions promoted uniformity in the information gathered, and made it simpler to compare and evaluate results.
3. ***Anonymity and Confidentiality***: Participants remained anonymous, which promoted open and honest feedback. It safeguarded sensitive information, and confidentiality measures were put in place.
4. ***Time and cost efficiency***: Surveys and questionnaires were given electronically, which saved time and money compared to performing one-on-one interviews. Additionally, they were passed on without the need for travel to a geographically spread audience.
5. ***Quantitative Data Analysis***: Surveys produced quantitative data that was analyzed statistically to find trends, patterns, and correlations.

**Disadvantages**

1. ***Limited in depth***: Questions and surveys provided respondents with limited information to elaborate on their answers or provide context.
2. ***Response bias***: Respondents may have had different meanings or interpreted questions differently, resulting in response bias or inaccuracies in the data collected.
3. ***Low response***: There was a risk of low response rates, which affected the representativeness and validity of the data collected. Some participants did not participate, identifying potential sources of bias.
4. ***Lack of Flexibility***: Surveys have predetermined questions and answers, which limited the ability to pick up on confusion or unexpected topics that may have arisen in open discussion.
5. ***Failure to clarify ambiguities***: Unlike interviews or focus groups, there was no opportunity for immediate clarification or follow-up questions to resolve ambiguities there not fully explained in the answer.

### **3.3.2 Interviews**

One-on-one interviews will be conducted with stakeholders, users and subject matter experts to gather information about their needs, expectations and pain points. We, as the project team will prepare structured and semi-structured questions to guide the interview process.

**Advantages of using Interviews:**

1. ***Automated data collection***: Interviews provided an opportunity to engage directly with stakeholders, users, or subject matter experts (SMEs). It provided team members with first-hand information about system requirements, user requirements, and specific information that may not be available in written documentation.
2. ***Clarifies ambiguity***: Interviews allowed team members to ask questions and explore clarity in real time. This helped eliminate any ambiguity or misunderstanding of the project requirements, creating a clear understanding between the development team and stakeholders
3. ***Context Insight***: The interview provided a platform for understanding the context in which the program will be implemented. By talking to users or SMEs, team members gained insight into existing processes, business processes and challenges faced by users. This context was valuable for planning to better meet the specific needs of users.
4. ***Building Relationships***: Conducting interviews enabled participants to build relationships with stakeholders. This contributed to better collaboration throughout the project and built trust between the development team and stakeholders.

**Disadvantages of Using Interviews:**

1. ***Time-consuming***: Conducting interviews was a time-consuming process, especially due to dealing with a large number of users. Planning, preparing questionnaires, interviewing, and analyzing data collected was time-consuming, potentially delaying the project timeline
2. ***Limited perspective***: Interviews are subjective and depend on the opinions, experiences and knowledge of the respondents. This led to biases or limited opinions. It may have neglected certain aspects or failed to capture the full range of user needs, resulting in incomplete or inaccurate requirements
3. ***Cost***: Interviews incurred a travel expense, as some participants and users were geographically dispersed. This increased the overall cost of the project, because when compared to other fact-finding methods, most of which do not require physical presence.
4. ***Miscommunication and misinterpretation***: Miscommunication or misinterpretation may have occurred during interviews, leading to misunderstandings or misconceptions. Interviewers and interviewees may have had different interpretations of certain terms or concepts, which may have affected the validity of the data collected.

## **Feasibility Study**

### **Technical Feasibility**

The technical feasibility study assessed whether the required technology infrastructure, hardware and software were available for use or could be developed.

Reasons for conducting this study included:

1. ***Compatibility***- Determined if the existing technology infrastructure could support the bus ticketing system's functionalities, such as ticket booking, seat selection, payment processing, and real-time data updates. Ensuring compatibility avoids costly infrastructure upgrades or replacements.
2. ***Development capabilities***- Assessed whether the development team had the necessary technical skills and expertise to build the ticketing system or if additional resources, such as specialized software or hardware, were required.
3. ***Scalability***- Evaluated if the system could handle increasing user demand and accommodate future growth. This consideration was crucial for ensuring long-term sustainability and avoiding performance bottlenecks.

### **Operational Feasibility**

The operational feasibility study assesses whether the bus ticketing system can be effectively integrated into existing operational processes.

Reasons for conducting this study included:

1. ***Stakeholder acceptance***- Determined the willingness and readiness of the Super Metro bus operators, staff, and passengers to adopt and use the ticketing system. Assessed the impact on existing workflows and identified potential resistance or challenges that may arise during implementation.
2. ***Process integration***- Evaluated how the ticketing system would fit into the existing bus operations, including ticket issuance, boarding, seat management, and reporting. Assessed the level of disruption and the feasibility of incorporating the system seamlessly into daily operations.
3. ***User experience***- Analyzed how the system would improve the overall passenger experience. Considered factors such as ease of use, efficiency, reliability, and convenience. Ensured that the system enhances passenger satisfaction and provides a competitive advantage.

### **Legal and Regulatory Feasibility**

The legal and regulatory feasibility study examined the compliance requirements and potential legal challenges associated with implementing a bus ticketing system.

Reasons for conducting this study included:

1. ***Data protection and privacy***- Ensured that the system complied with applicable data protection and privacy laws. Assessed how customer data would be collected, stored, processed, and secured.
2. ***Payment regulations***- Evaluated compliance with financial regulations and standards for payment processing. Considered encryption, fraud prevention measures and adherence to local payment regulations.
3. ***Ticketing regulations***- Understood and complied with local, regional or national regulations related to ticketing, fare structures, refund policies, and passenger rights. Ensured that the system aligned with regulatory requirements to avoid legal complications or penalties.

### **Social Feasibility**

The social feasibility of the Local Service Search Engine Management System refers to its compatibility with social and cultural norms, its potential impact on society, and its ability to address social needs and concerns. In the context of this system, several factors contribute to its social feasibility:

1. ***Community Engagement***: The system facilitates community engagement by connecting local service providers with service seekers within their neighborhoods or communities. By promoting local businesses and fostering community interactions, the system contributes to the social fabric of the community.
2. ***Accessibility and Inclusivity***: The system aims to be accessible and inclusive to all members of society, including those with diverse backgrounds, abilities, and needs. It provides a platform for service seekers to find local services tailored to their specific requirements, promoting inclusivity and equal access to services.
3. ***Empowerment of Local Entrepreneurs***: The system empowers local entrepreneurs and small businesses by providing them with a platform to showcase their services and reach a wider audience. This can have a positive social impact by fostering entrepreneurship, creating job opportunities, and supporting local economies.
4. ***Trust and Reputation Building***: The system incorporates features such as user ratings, reviews, and verification mechanisms to build trust and credibility among users. By facilitating transparent and reliable interactions between service providers and seekers, the system helps to establish and maintain trust within the community.
5. ***Social Interaction and Networking***: The system facilitates social interaction and networking among users by enabling communication, collaboration, and feedback exchange. This can lead to the formation of social connections and networks within the community, enhancing social cohesion and community resilience.