

# Urban Clap Clone Web App Scope and Technical Solution: **The MVP**

---

Presented to you by:



We Deliver  
#DigitalSuccess

---

**Document Name:** Urban Clap Clone Web App Scope and Technical Solution: The MVP

**Document Type:** Proposal

**Document Version:** 1.A

**Author:** Deborishi Ganguly

**Product Owner:** Sreya Mukherjee

**Sponsor:** Client

## Table of Contents

1. Introduction .....	4
1.1 Purpose .....	4
1.2 Business Success Criteria and benefits of the planned implementation .....	4
1.3 Market Research .....	5
2. Phase Scope and Platform Selection.....	5
3. The Solution for the MVP .....	5
3.1 Technical Aspects .....	5
3.2 Roles .....	6
3.3 UI/UX .....	6
3.4 Key Functional Features .....	6
3.5 Solution Stack .....	8
3.4.1 Cloud Stack .....	9
3.6 Hosting.....	9
3.7 Database .....	10
3.8 Non-functional Requirements .....	11
3.8.1 Security.....	11
3.8.2 Performance - Load Optimisation.....	11
3.8.3 Portability - Mobile Responsiveness .....	12
3.8.3 Localization .....	12
3.9 Web Application Architecture .....	12
3.10 Business Process Modeling .....	14
3.11 Content .....	15
4. Version Control .....	15
5. Testing .....	15
5.1 Functional Testing.....	15
5.2 User Acceptance Trial.....	15
5.3 Bug Tracking .....	15
6. Software Development Methodology.....	16
7. Change Management .....	17
8. Documentation .....	17
9. Warranty.....	17

10. Time Estimation & Resource Implication .....	18
1.A – System Analysis.....	18
1.B – Application Design.....	18
1.C – Application Development.....	18
1.D – Software Testing.....	18
1.E – Database setup and administration .....	19
1.F – Cloud Administration for Deployment .....	19
11. Cost Estimation Analysis .....	19
12. Contract .....	20
13. Proposal Acceptance .....	20
14. Glossary and Index.....	20
15. Terms and Conditions.....	20

## 1. Introduction

### 1.1 Purpose

As per the telephonic conversation on the 20<sup>th</sup> of July, 2020, the client wanted a Minimum Viable Product (MVP) to showcase the most fundamental functionalities of an on-demand, [Urban Clap](#) like clone software, to attract Venture Capitalists and Angel funders. The logic is to start simple and scale up functionalities as the bootstrap startup secures more funding. To meet this purpose, the software is proposed to be a simple web application.

Our proposal values speed, cost, design, functionality, and durability of the project and this document serves to showcase how the solution can minimize trade-offs vis-à-vis each value.

### 1.2 Business Success Criteria and benefits of the planned implementation

Below is the primary objective of this phase:

- Attract VC/Angel funding

The following are the immediate benefits that can be foreseen with the intended implementation:

- 1) The solution ensures that the prototype has minimum functionality with the maximum performance
- 2) The solution can be released in market for the 2<sup>nd</sup> phase of the project where brand awareness market acceptance and customer feedback can be gauged

## 1.3 Market Research

While on-demand service mobile applications have gained immense market acceptance in large metropolitan cities, smaller cities and town have not seen the same level of adoption. Hyperlocal services in this places too need to expand and modernize, to expand the area of operations in urban and suburban settings. The app can be easily marketed in these areas.

## 2. Phase Scope and Platform Selection

This the 1<sup>st</sup> phase of the project and it shall deal with making a software application capable of capturing end user details for registration, registration and of listing service providers and the application collecting payments for service dispatch.

We propose a **web application** in place of a mobile app as we do not expect usage of the app to take place more than **1 to 2 times** in a given day, in this initial phase, if it is to be operationalized.

## 3. The Solution for the MVP

### 3.1 Technical Aspects

To speed up the development of the application, there are a total of 5 technical aspects that will be either executed concurrently or ad hoc. They are not sequential.

- 1) Prototyping the solution (with Wireframing through MS Visio and mock-ups)
- 2) Front end design
- 3) Development and functional implementation
- 4) Testing

Maintenance will be offered as through an **Annual Maintenance Contract (AMC)**.

## 3.2 Roles

The following are the **roles** for the web application being developed:

- 1) **Super Administrator or Admin** – the client or his/her representative , who has holistic control over the entire operation of the web app, overview of its health and the revenue stream
- 2) **Service Provider** – B2C/B2B entities that offer services to the end users
- 3) **Customer** – the end user who would consume the services being provided by the service providers, who are either individuals or other businesses

## 3.3 UI/UX

The application is to have a simple User Interface (UI) that is meant to be intuitive from the first look.

However, since the MVP is not a prototype, it must have a degree of appeal for early adoption. Here are some of the action items for producing UI:

- Outline an user flow – map how users will move through the application
- To capture the brand requirements, we would provide a **UI/UX questionnaire** which would capture the basic brand aesthetics.
- Prototyping the MVP will see both wireframes and mock ups produced.
- Finally, the mock-ups will be converted into actionable UI.
- The UI designed will be iterative

## 3.4 Key Functional Features

The following is a high level overview of the functionalities that will be accorded through the software:

## 1) Customers Features

- Registration & Login module – through either email, with verification, or social media
  - Ability to reset password
- Browse and view Service Provider & Price Ranges
- Service Search Filters
- Single Payment Option – through Paypal (owing to its popularity and ease of use)
- Call service provider – (Phone Number is Call-to-action)
- Place Service Request
- Cancel Service Request

## 2) Service Providers Features

- Registration & Login module – through email, with verification
  - Ability to reset password
- Dashboard with ability to –  
View/Add/Delete/Update Name, Phone Number  
  
View/Add/Delete/Update Service Rate & Charges Listing- Hourly, Fixed, Experienced Base Quote  
  
View/Add/Delete/Update Set Availability & Service Times
- Accept/Reject Service Request from customer
- View Customer contact and address details for active service requests
- View status of job request Real-time (Active, Cancelled)

### 3) Admin Panel Features

- Login through email
  - Ability to reset password
- Ability to view/add/delete Customer and Service provider profiles
- Ability to change user IDs and passwords for Customers and Service providers
- Access to Admin dashboard
- Content Management System – ability to add/delete/update text, images and videos
- Record Management System For customers & Service providers
- Work Records, Bill & Payment History – (captured for future analytics)

### 3.5 Solution Stack

We are proposing the use of the MERN stack not only for the best user experience but also keeping an eye on the future where this web application can be converted into a Progressive Web Application for deployment on Google Playstore for Android and on the Apple App store.

The MERN stack is a software stack that includes four open-source technologies: (MongoDB, Express.js, React, and Node.js). These components provide an end-to-end framework for building dynamic web sites and web applications.

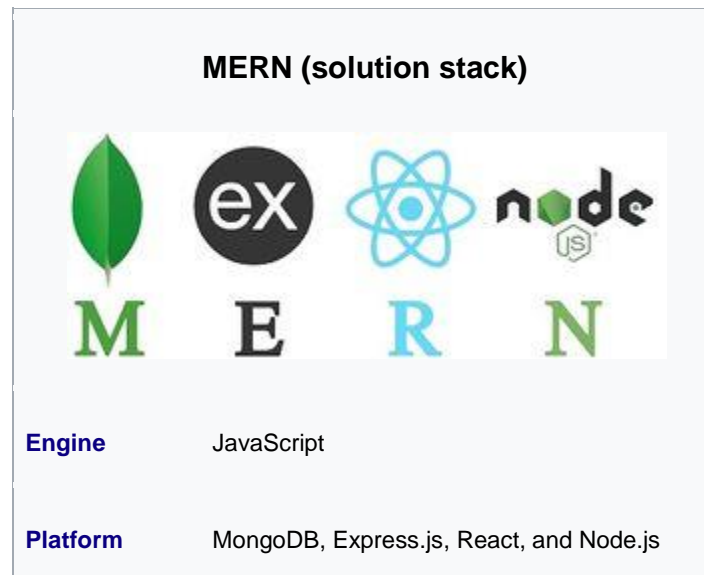
- MongoDB is a database system
- Node.js is a server-side runtime environment
- Express.js is a web framework for Node.js and it allows the implementation of web application features
- React is a client-side JavaScript library used for building user interfaces. For the front-end, it is **fast** and it is **light**, which goes towards **reducing** the cost of hosting this application



### 3.4.1 Cloud Stack

Part of the solution stack is based in the AWS cloud and this sub-section aims to cover them in brief.

- The application server will be the Linux based Ubuntu, running on an [Amazon EC2](#) VM instance
- The [web server](#) used will be a combination of [Nginx](#), [Node.js](#) and [MongoDB](#)
- As the solution needs hosting, the storage will be on [Amazon EBS](#) – discussed further in the hosting section
- Optionally, a web based firewall can applied to the solution through [Amazon WAF](#) once the application gains more users



*Diagram 1: The MERN Stack*

### 3.6 Hosting

To keep initial costs down, the web app will be hosted on the free tier of a t1.micro [Amazon EC2](#) instance, which entails a 750 hours of usage of an Ubuntu instance.

In the table below are the technical specifications of the t1.micro instance:

Instance Family	Instance Type	Processor Arch	vCPU	Memory (GiB)	Instance Storage (GB)	EBS-optimized Available	Network Performance
Micro	t1.micro	32-bit or 64-bit	1	0.613	EBS Only (30 GB of Storage)	-	Very Low

**Table 1:** The t1.micro instance

The benefits of cloud hosting are the following:

- 1) High load through usage of the app can be assuaged by automated scaling up of networking and computing instances.
- 2) 3rd Party providers guarantee a high resilience to the uptimes of the service
- 3) Availability of Platform as a Service (PaaS), a common offering among these providers, ensures that the cloud providers would take care of the hardware, networking, databases, platforms, load balancing and security etc. without constant manual intervention, reducing the requirement of manual resources.
- 4) There's extreme flexibility in the pricing, it is calculated as per usage of features and number of times a feature is used (pay-as-you-go).
- 5) Redundancy in safety of data as data can be backed up in multiple data centres around the country.
- 6) Data and network security is given priority and a guarantee is given for it's protection.

Beyond this usage, the projected price of hosting can be found in the payment breakup section below.

### 3.7 Database

We are proposing the usage of a NoSQL Database Management System, MongoDB.

Below are some benefits of MongoDB:

- Document Oriented Storage – Data is stored in the form of JSON style documents.

- Index on any attribute
- Replication and high availability
- Auto-Sharding
- Rich queries
- Fast in-place updates
- Professional support by MongoDB

### 3.8 Non-functional Requirements

There are also technical specifications which do not provide any functionality to the end user however is essential to the overall usability, resilience and health of the application. While reliability, maintainability and availability are accorded by the AWS cloud hosting, listed below are few other aspects.

#### 3.8.1 Security

The AWS Cloud hosting accords the solution strong security and it is unnecessary for the solution to be further secured through a Web Application Firewall (WAF).

The IT infrastructure that AWS provides to its customers is designed and managed in alignment with best security practices and a variety of IT security standards. The following is a partial list of assurance programs with which AWS complies:

- SOC 1/ISAE 3402, SOC 2, SOC 3
- FISMA, DIACAP, and FedRAMP
- PCI DSS Level 1
- ISO 9001, ISO 27001, ISO 27017, ISO 27018

#### 3.8.2 Performance - Load Optimisation

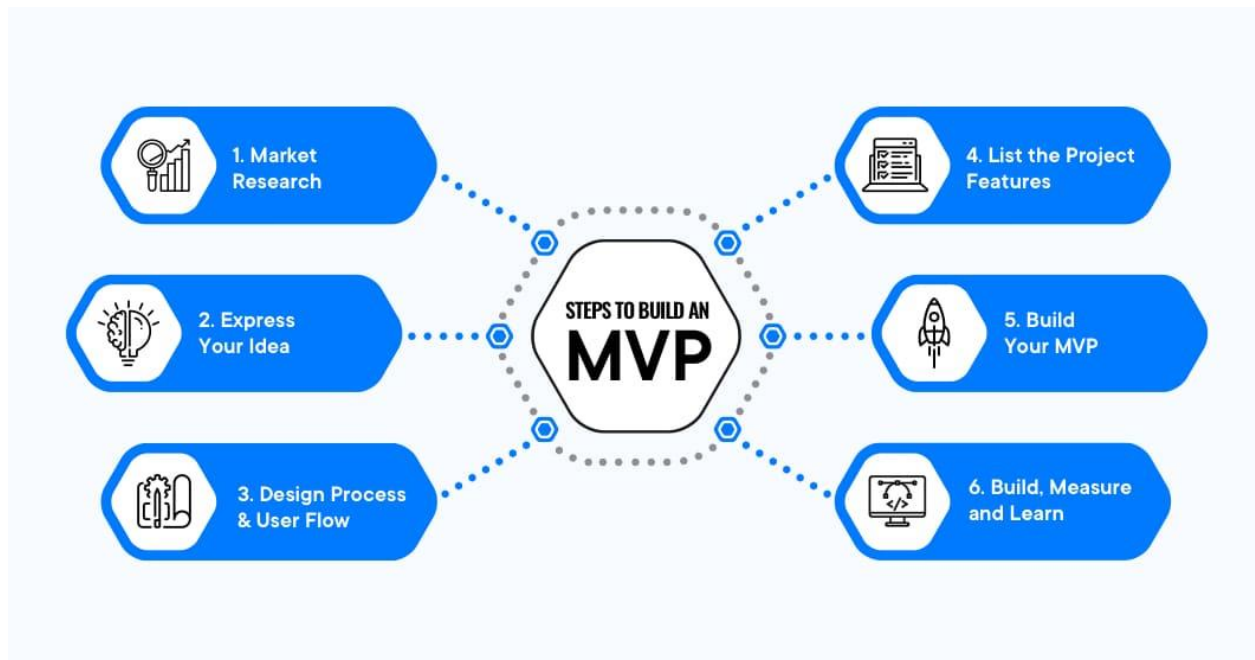
The usage of the MERN stack accords industry standard load times of around 5 seconds.

### 3.8.3 Portability - Mobile Responsiveness

To demonstrate the operability of the application on a mobile device, the app will be mobile responsive.

### 3.8.3 Localization

This app aims to be hyperlocal. Content and services will be local for target areas.



*Diagram 2: A Recap of the MVP building Process*

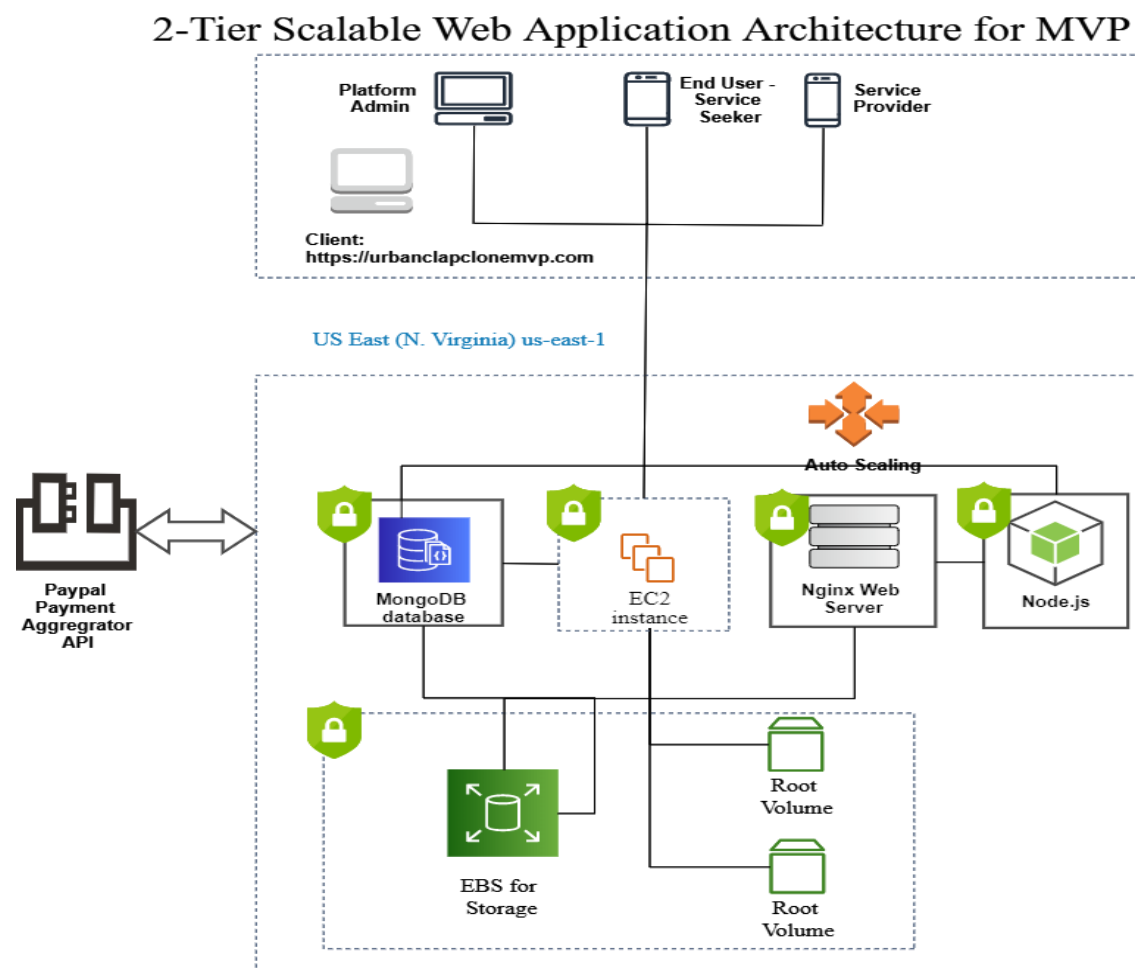
### 3.9 Web Application Architecture

To attract the best kind of investment, the app must be resilient and light for the user experience (UX).

To ensure that the app is fast, we propose the usage of 2-tier web architecture. Since the app is hosted on the AWS cloud, it can easily be converted into 3-tier architecture or microservices based application when the time comes to scale up.

Furthermore, due to the robust security of the AWS cloud architecture, many security risks are mitigated.

The following diagram illustrates the architecture of the web app:



*Diagram 3: 2-Tier Scalable Web Application Architecture for the MVP*

### 3.10 Business Process Modeling

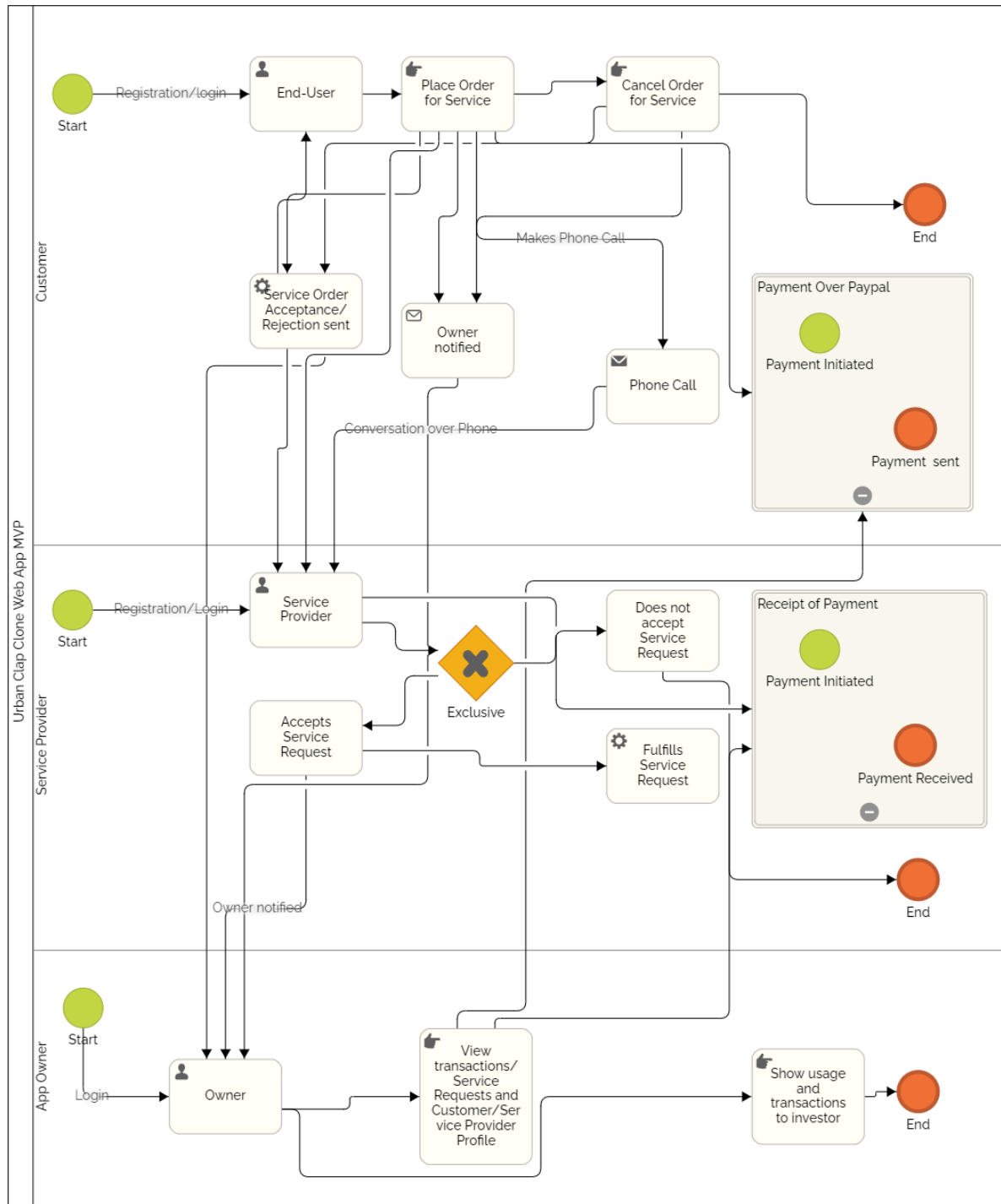


Diagram 4: BPMN of MVP app Operations

### **3.11 Content**

Content of the web application will be supplied by the client. If content writing is required, a content writer can be engaged at an additional cost of USD 10 per hour.

## **4. Version Control**

All resources would collaboratively store their work on a Github private repository.

## **5. Testing**

### **5.1 Functional Testing**

For the functional testing of the web application - [Selenium](#) Webdriver, an automation framework, will be used.

### **5.2 User Acceptance Trial**

Following the functional test, an User Acceptance Trial (UAT) will be conducted for the client.

An UAT document will be provided to the client to assess the functionality of the application.

### **5.3 Bug Tracking**

Mantis BT will be used for bug tracking.

## 6. Software Development Methodology

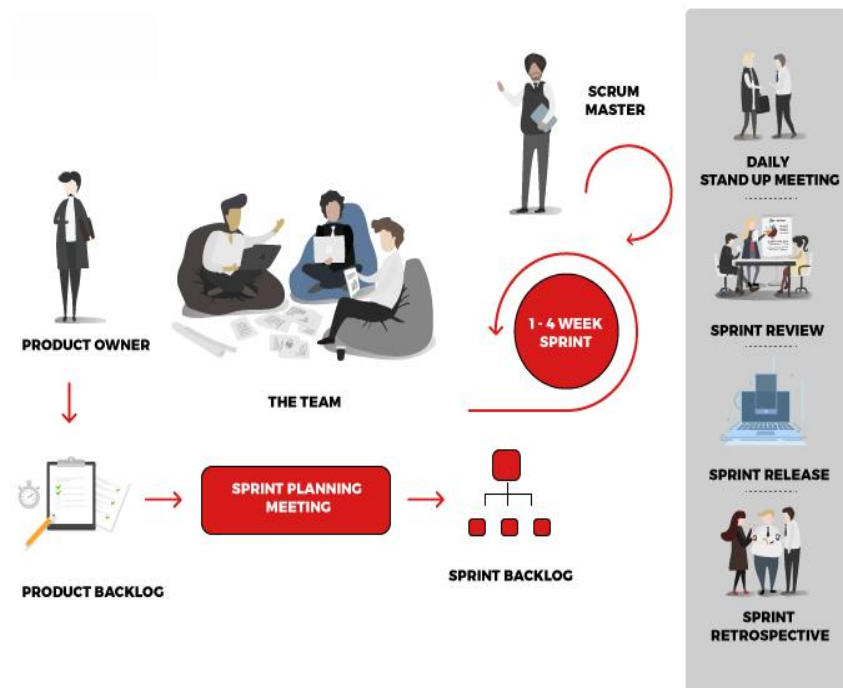
Our solution aims to be iterative, incremental and evolutionary to ensure that the app is ready for the next phase. The Minimum Viable Product is the basic for [Agile methodology](#).

To reduce the risk of building this MVP, Agile methodology is going to be used for this project.

Benefits of Agile Methodology:

- reduce risks;
- be more adaptable to changes, and (most importantly),
- build a product that will more likely succeed in the market.

It presupposes dividing the project into smaller parts called iterations. The result of each iteration is a working piece of software that allows collecting the feedback, analyzing it and moving forward.



*Diagram 5: The Agile Scrum Framework ecosystem*



## 7. Change Management

As an Agile Software Development company, we welcome changes to the requirement, however, if changes are sought, they will be considered a '**change order**'. There would be a subsequent change in pricing as per the effort hours.

Any request for a change order, can be placed through a change order form provided by the Account /Relationship Manager.

## 8. Documentation

During the course of the project, the following documentation would be provided from our end:

- 1) Project Plan – from Project Manager and Business Analyst
- 2) Risk Management Matrix document
- 3) UI/UX Questionnaire – from UI Designer
- 4) UAT check list – from Tester
- 5) Application Manual – from Project Manager and Developer
- 6) Annual Maintenance Contract – from Sales or Account Manager

Certain documents will need input from the client and timelines can be adversely impacted if required inputs are not timely given.

## 9. Warranty

From the date of acceptance, the app will be under warranty for a period of 3 months and it will be the responsibility of IndusNet Technologies to fix all bugs to ensure the smooth running of the application.

Also, within this period of time if the application is hacked or infected with malware, we will take the responsibility of securing the application.

IndusNet Technologies will not be liable for the loss of any customer or service provider data in the event of any intrusion or malware infections.

Beyond these 3 months, an extension of the warranty has a separate cost for which an **Annual Maintenance Contract** can be signed.

## 10. Time Estimation & Resource Implication

Time estimation is calculated as per the full business day, which consists of 8 hours of work.

### 1.A – System Analysis

Resource	Time Estimate	No. of resources
Business Analyst	5 business days	1
Project Manager	10 business days	1

### 1.B – Application Design

Resource	Time Estimate	No. of resources
UI/UX Designer	7 business days	1

Resource	Time Estimate	No. of resources
React Front-end Developer	10 business days	1

### 1.C – Application Development

Resource	Time Estimate	No. of resources
Express/Node.js Developer	25 business days	1

### 1.D – Software Testing

Resource	Time Estimate	No. of resources
Software Tester	5 business days	1

#### 1.E – Database setup and administration

Resource	Time Estimate	No. of resources
MongoDB Administrator	10 business days	1

#### 1.F – Cloud Administration for Deployment

Resource	Time Estimate	No. of resources
AWS Cloud Administrator	2 business days	1

Some of the above roles will also write documentation and the effort is calculated into the time estimate.

Many of the processes are concurrent. The time estimation for the duration of the project is a range of 45 to 60 days.

## 11. Cost Estimation Analysis

We propose an Agile Fixed Price model for the MVP software development.

Resource	Cost per resource per hour in US Dollars	Total cost of resources in US Dollars
Business Analyst	15	600
Project Manager	15	1200
UI/UX Designer	15	840
React Front-end Developer	15	840
Express/Node.js Developer	20	4000
Software Tester	15	600
MongoDB Administrator	20	1600
AWS Cloud Administrator	30	480

Estimated Cost of Development	10,160
-------------------------------	--------

## 12. Contract

Upon acceptance of this proposal, the contract document will be signed by both IndusNet Technologies and the client.

## 13. Proposal Acceptance

This section is to capture the acceptance of this document. VeriSign e-signature is recommended.

-----

Name of Client:

Place of Signing:

Date:

## 14. Glossary and Index

Glossary of IT terms, components and services.

## 15. Terms and Conditions

IndusNet Technologies T&C

