**MAIN FEATURES – SCOPE OF PROJECT**

**Important:** *Make a video presentation to investors and will be uploaded to YouTube and our social media accounts and to our site.*

* **Social Network (Core)**
  + **Viewing**
  + **Posting**
  + **Follow**
  + **Listing**
  + **Reviews**
  + **Referral**
* **Messaging (Core)**
  + **Chat – Client and Server**
  + *Voice – Support in the Future*
* **Courier Service (Beta)**
  + **Navigation Page with Google Direction API**
  + **Messaging Page for active transaction.**
  + **Transaction Page w/ Summary Dashboard**
  + **Document and Verification**
* **Wallet System (Beta)**
  + **PAYPAL**
  + **GCASH**
  + *Paymaya – Support in the Future*
* **Market / Grocery (Beta)**
  + **Inhouse Grocery tab**
  + **Seller tab view by category**
* eCommerce (Stable) – 21DAY to 1MONTH
  + Store Browser
  + Store Front
  + Product by Category
  + Order List
  + Cart View
  + Checkout
  + Payment
  + Complete
  + Review Page
  + Tracking Page
  + Algorithm – Insert to Waiting for Rider.
* Point of Sale (Stable) - 21DAY to 1MONTH
  + Dashboard – Summary
  + Messaging
  + Product List
  + Category List
  + Transactions
  + Voucher System
  + Documents and Verification

**BETA RELEASE – CONTROLLED PRODUCTION**

**Project Motion Graphics and Presentation to Investors.** – **7 DAY**

* Writing to Approval of initial Video script and visualization. 2DAY
* Video editing and sound scoring - 3DAY
* Minor adjustments or revisions of finish video. – 2DAY

**WORDPRESS PROJECT WELCOME SITE – 5 DAY**

* Complete all company specific illustration assets. – 2 DAY
* Complete and Write content to all the parts of site. – 2DAY
* Minor adjustments and bug fixing to quality control of the implementation. – 1DAY

**AUTHENTICATION AND REGISTRATION – 12 DAYS**

* Implementing our Native signin, signup, forgot with email notification. **– 5 DAYS**
* Write the backend RestAPI for signin and implement on client, using username and password and return token to the user. – 1DAY (Done)
* Write the backend RestAPI for signup with email notification using our own official email to send activation key to the given user email then handle password reset on client. – 2DAY
* Write the backend RestAPI for forgot password that will send new password reset activation key to the user email and handle password reset on client. – 1DAY
* Minor adjustments and bug fixing to quality control of the implementation. – 1DAY
* Implementing Facebook SDK to use Facebook login->signup. **– 4 DAYS**
* Application for API access and setting up client configuration. – 1DAY
* Write a code that will bind Facebook token to our native **signin** and **signup**. – 2DAY
* Minor adjustments and bug fixing to quality control of the implementation. – 1DAY
* Implementing Google SDK to use Google login->signup. – **4 DAYS**
* Application for API access and setting up client configuration. – 1DAY
* Write a code that will bind Facebook token to our native **signin** and **signup**. – 2DAY
* Minor adjustments and bug fixing to quality control of the implementation. – 1DAY

**DYNAMIC DATA SYNCHRONIZATION (Social Network) – 5 DAYS**

* On success user signin or complete signup, the application will automatically ask for primary user information like name, photo, banner, etc. 2DAY (Done)
* On Home page, Write a backend restapi that will initially request for 12 post entry and then dynamically increment 5 post entry if user is near at the end of post. Next to the, implement the displaying of this data on the client side. – 2DAY
* On Profile page (either local or other), Write a backend restapi that will initially request for 12 post entry and then dynamically increment 5 post entry if user is near at the end of post. Next to the, implement the displaying of this data on the client side. – 2DAY
* On Profile and Home, make a simple system that will distinguish what type of post will be display on the feed. Every type should be easily distinguishable. – 1DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY

**USER POST ENTRY TO SOCIAL NETWORK FEED – 7 DAYS**

* Either on Home or Profile, write a backend RestApi for inserting new post type of status, then implement it on the client side, user should display a modal popup editor for showing Status post. – 2 DAY
* Either on Home or Profile, write a backend RestApi for inserting new post type of selling, then implement it on the client side, user should display a modal popup editor for showing Selling of post. – 2 DAY
* Either on Home or Profile, write a backend RestApi for inserting new post type of request, then implement it on the client side, user should display a modal popup editor for showing Request of post. – 2 DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY

**USER CONNECTIONS OR NETWORKING – 5DAY**

* Design and Develop a Database for handling this user networking. – 1DAY
* Make a RestAPI for following, unfollowing, a user from another user. -1DAY
* Implement on the client side the display of the reviews to profile and other profile page. -2DAY
* Quality control and checking for bugs or any unhandled exception. – 1DAY

**REVIEW SYSTEM WITH SELECTIVE CHECK – 6DAY**

* Design and Develop a Database for handling this wallet system. – 1DAY
* Make a simple backend ui for checking review system log. -1DAY
* Make a RestAPI for inserting, deleting review log on the system. -1DAY
* Implement on the client side the display of the reviews to profile and other profile page. -2DAY
* Quality control and checking for bugs or any unhandled exception. – 1DAY

**WALLET SYSTEM WITH TWO CURRENCY HANDLER – 7DAY**

* Design and Develop a Database for handling this wallet system. – 1DAY
* Make a simple backend ui for checking wallet system log. -1DAY
* Make a RestAPI for inserting, moving currency to other user. -2DAY
* Implement on the client side the display of the logs and transaction as well as the summary of logs and visualization. -2DAY
* Quality control and checking for bugs or any unhandled exception. – 1DAY

**REFERAL SYSTEM WITH USER {CREDITING} – 6DAY**

* Design and Develop a Database for handling this referral system. – 1DAY
* Make a simple backend ui for checking wallet system log. -1DAY
* Write a WordPress routing override for short link of referral and perform the crediting mechanism for this referral or whatever to implement. -2DAY
* Implement on the client side for showing a button or ui for sharing referral. -1DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY

**CHAT MESSAGING IMPLEMENTATION ON CLIENT – 10DAY**

* Evaluate and finalize the server of choice that we will use as our chat backend, it should be real-time and with pay as you go feature. USocketNet! – 1DAY
* Quickly, make a production ready build for USocketNet that will only utilized the messaging module of the project. – 2DAY
* We will be adding a tab for Request messages with active or inactive status. -1DAY
* Publish a USocketNet server on the cloud from the previously stabilized build. -1DAY
* Design and Develop an efficient action flow for sending and receiving realtime data from user to user. Database: usn\_message – 1DAY
* Make a RestAPI that will post the message to wordpress from a success message callaback. – 1DAY
* Implement on the client that will send a message to a specific wp user by ID and return a callback. – 2DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY

**COURIER NAVIGATION VIA POST REQUEST – 8DAY**

* Design and Develop a backend for handling this navigation system. – 1DAY
* From transaction page, after the user click a transaction to accept, the navigation page will show current direction view. -1DAY
* Implement multiple dropoff direction api and ask google for estimated time of arrival based on current and previous speed. – 2DAY
* User be able to show info about the request just by pressing (i) logo on the top upper side of the page. – 1DAY
* User can Cancel and Complete the current active transactions with tabview on the different active transaction. -2DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY
* **PLANNING:** Rider will send its geolocation on USocketNet to be able for the client to request for current GEOLOCATION of the rider. Like what the Lalamove implemented.

**COURIER MESSAGING SPECIFIC TO ACTIVE TRANSACTION – 5DAY**

* Design and Develop a backend for handling this messaging system. – 1DAY
* From the transaction page, messaging will the id of that user to have an active conversation on the chat page. -1DAY
* Implement on the client side on sending and receiving messages. -2DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY

**COURIER TRANSACTION LISTING WITH TABVIEW – 5DAY**

* Design and Develop a backend for handling this messaging system. – 1DAY
* On newsfeed accept success, that request will be forward to this transaction viewer and will be set as active. – 2DAY
* On transaction completed or cancelled, this transaction will be tagged as inactive and set its status either Complete or Cancelled. – 1DAY
* The rider will be able to select one transaction to preview on navigation page. - 1DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY

**COURIER DOCUMENT AND VERIFICATION PAGE – 8DAY**

* Design and Develop a backend for handling this verification system. – 1DAY
* User will be able to apply for verification and upload list of required and optional data. – 2DAY
* Make a backend ui for accepting and comment the user about the status or issue. -2DAY
* User will be able to received comments and reply to comments about this. -2DAY
* Quality control and checking for bugs or any unhandled exception. – 1DAY

**MARKET INHOUSE GROCERY LISTING – 7 DAY**

* Design and Develop a backend for handling this messaging system. – 1DAY
* Make a backend ui for adding, editing, removing grocery item. – 1DAY
* Write a RestAPI to get the category of this post type and then fetch by category. -2DAY
* Implement displaying of the grocery item on market tab on the client side. -2DAY
* Quality control and checking for bugs or any unhandled exception. – 1DAY

**SELLER ITEM VIEW ON MARKET – 4 DAY**

* Display the seller item per category on the client side, but fist write a RestAPI to be fetch by client on request for category list and fetch item by category id. – 3DAY
* Quality control and checking for bugs or any unhandled exception. – 1DAY

**PAYMENT GATEWAY WITH PAYPAL, GCASH, AND PAYMAYA – 10DAY**

* PAYPAL – From application to implementing of sdk to the app. Making RestAPI to server and the client to process current transaction. - 4DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY
* GCASH - From application to implementing of sdk to the app. Making RestAPI to server and the client to process current transaction. – 4DAY
* Quality control and checking for bugs or any unhandled exception. – 1 DAY

**Building Publishing of Android apk and IOS Package** – **7 DAYS**

* Writing to publishing content for our developer PlayStore and AppStore. – 1Day
* Writing content for Android and IOS like description and capturing Previews. – 2Day
* Publishing on PlayStore with all related Meta binding to related sites. - 0.5Day
* Publishing on AppStore with all related Meta binding to related sites. – 0.5Day
* Quality control and double checking all data published and setting linkage to our site. – 3Days
  + Make sure SignUp and SignIn is totally working.
  + Make sure Facebook login is working properly.
  + Make sure Google login is working properly.
  + Make sure Firebase analytics is working properly.
  + Monitor crash reports on Firebase crashlytics.

**STABLE RELEASE – v1.0.0**

QUALITY CONTROL AND TESTING

**Installation testing – 1DAY**

Once the application is ready, tests need to conduct installation testing to ensure that the user can smoothly install or uninstall the application. Additionally, they also have to check that the application is updating properly and does not crash when upgrading from an older version to a newer one. Testers also have to ensure that all application data is completely removed when an application is uninstalled.

**Target Device and OS testing – 1DAY**

Mobile testers have to ensure that the mobile app functions as designed across a plethora of mobile devices and operating systems. Using real devices and device simulators testers, they can check the basic application functionality and understand the application behavior across the selected devices and form factors. Applications also have to be tested across all major OS versions in the present installed base to ensure that it performs as designed irrespective of the operating system.

**UI and UX testing – 1DAY**

UI and UX testing are essential to test the look and feel of the application. This testing has to be done from the users’ perspective to ensure that the application is intuitive, easy to use, and has industry-accepted interfaces. Testing is needed to ensure that language- translation facilities are available, menus and icons display correctly, and that the application items are synchronized with user actions.

**Functionality Testing – 1DAY**

Functionality testing tests the functional behavior of the application to ensure that the application is working according to the specified requirements. This involves testing user interactions and transactions to validate if all mandatory fields are working as designed. Testing is also needed to verify that the device is able to multitask and process requirements across platforms and devices when the app is being accessed. Since functional testing is quite comprehensive, testing teams may have to leverage test automation to increase coverage and efficiency for best results.

**Interrupt testing – 1DAY**

Users can be interrupted with calls, SMS, MMS, messages, notifications, network outage, device power cycle notification etc. when using an application. Mobile app testers have to perform interruption testing to ensure that the mobile app can capably handle these interruptions by going into a suspended state and then resuming functions once the interruptions are over. Testers can use monkey tools to generate multiple possible interrupts and look out for app crashes, freezes, UI glitches, battery consumption etc. and ensure that the app resumes the current view post the interruptions.

**Data network testing – 1DAY**

To provide useful functionalities, mobile apps rely on network connectivity. Conducting network simulation tests to simulate cellular networks for bandwidth issues to identify connectivity problems and bottlenecks and then study their impact on application performance fall under the purview of network testing. Testers have to ensure that the mobile app performs optimally with varying network speeds and is able to handle network transitions with ease.

**Hardware keys testing – 1DAY**

Mobile apps are packed with different hardware and sensors that can be used by the app. Gyroscope sensors, proximity sensors, location sensors, touchless sensors, ambient light sensors etc. and hardware features such as camera, storage, microphone, display etc. all can be used within the application itself. Mobile testers thus, have to test the mobile app in different sensor specific and hardware specific environments to enhance application performance.

**Performance Testing – 1DAY**

The objective of performance testing is to ensure that the mobile application is performing optimally understated performance requirements. Performance testing involves the testing of load conditions, network coverage support, and identification of application and infrastructure bottlenecks, response time, memory leaks, and application performance when only intermittent phases of connectivity are required.

**Load testing – 1DAY**

Testers also have to test application performance in light of sudden traffic surges, and ensure that high loads and stress on the application does not cause it to crash. The aim of load testing is to assess the maximum number of simultaneous users the application can support without impacting performance and assess the applications dependability when there is a surge in the number of users.

**Security testing – 1DAY**

Security testing involves gathering all the information regarding the application and identifying threats and vulnerability for the application using static and dynamic analysis of mobile source code. Testers have to check and ensure that the applications data and network security functionalities are in line with the given guidelines and that the application is only using permissions that it needs.

**SUMMARY:** *Mobile application testing begins with developing a testing strategy and designing of the test plans. The added complexity of devices, OS’ and usage specific conditions adds a special burden on the software testing function to ensure the most usable and best-performing app. How have you gone about testing your mobile apps to achieve this end?*