



Project Estimate and Roadmap - Sample

Prepared By: Harsh Murari

January 17, 2018

OVERVIEW

Project

Develop Mobile Applications for iOS and Android platform for <Client Project>. Following are the high level requirements.

Requirements

- MVP Mobile Application - to be supported for Android and iOS platforms.
- Application should allow login, viewing devices on map, selecting devices and displaying real-time data on the app.
- Mobile Application has following screens: Splash screen, Login, Registration, Dashboard, Maps, Device Details, Device Data and Settings.
- After logging in, mobile application uses device's location services to determine current latitude/longitude pair and displays appropriately centered map.
- On map screen, should be able to see available devices and select/unselect a device to view its details.
- From device details page, we can route to device data page: which shows real-time (delayed) data from a server using web-sockets/SocketIO.
- Device data page shows a graph/chart of data being collected in near real time, it keeps refreshing every few seconds, chart can be scrolled back and forth etc.
- Mobile app needs to follow existing themes, color schemes, UI mockups and CSS assets provided.
- Mobile app integrates with user management and Auth APIs, device management and device real time data APIs provided.
- Mobile app image needs to be integrated with Jenkins/Gitlab CI to build and generate application image.
- For initial application, we will do manual tests for the application (no automated tests).
- Integrate with APIs provided by client for device management, user management and real-time device data.

In addition to the requirements above, following support roles will need to be addressed.

- Upload application to Android PlayStore and Apple AppStore. Support through initial app approval process, customer testing through TestFlight, and continuous security and feature updates of the application.
- Client will review the current mockups and share feedback. Client and Rhoynar team will go through couple of Design Scrum iterations to get down to a validated product prototype and then begin the implementation.
- Client will provide the infrastructure for device management and infrastructure for device data collection.
- Client will provide timely response for questions, clarifications and design concerns that Rhoynar team may have.

Project Details: Initial Mockups

<Product Mockups to be included separately from this file>

Project Details: Components, APIs and Framework Selection

Component	Frameworks, APIs and Other Integrations
React Native Template Application	<ul style="list-style-type: none"> - Using React Native Framework with Bootstrap CSS - Use provided mock-up assets and their CSS - Integrate with redux for component state management - Integrate with RxJs/Observables to handle notifications. - React-navigation for navigations and routing.
Splash Screen	<ul style="list-style-type: none"> - Use react-native-splash-screen
Registration Page/Login Page	<ul style="list-style-type: none"> - Perform client side validations for registration/login fields. - Need to interact with Registration/User Management APIs - http/REST integration (axios/react-native-rest-client) - Use auth service like passportjs - Implement Support Auth token/key management - Persona management
Dashboard Page (Map View)	<ul style="list-style-type: none"> - Shows a map view around current location. - Shows markers for different devices that can be selected/unselected. - Selected markers can enable streaming data from those devices. - Enable device location services to get current location - Use google maps APIs for maps display. - Map-markers: active/dormant, selected/unselected - Device state management showing state of selected map-marker/device. - Integrate with client provided Device management APIs - Support for getting devices nearby a given latlong - Get status (active/dormant) and details of device (name, type, details etc) - Support for State management/observables (redux/rxjs)



Device Details Page	<ul style="list-style-type: none">- Shows details of the device, button to get real-time data from device- Buttons can be grayed out if device is unavailable.- Use redux to manage state (which device selected etc.).- Integrate with device management APIs.- Get device details/status (active/dormant etc).- Use axios for REST APIs.
Device Data Stream Page	<ul style="list-style-type: none">- Display real-time data from those end-points.- Uses web-sockets to communicate between server and device.- Use react-native-charts to display data. Allow to scroll, pinch, zoom the plots.- Support refreshing charts/plots asynchronously using sockets.- Use react navigation to close screen and sockets etc.- Redux for state management- Uses Axios for REST APIs.
Settings Page	<ul style="list-style-type: none">- Shows User profile, settings, terms and conditions, notification and allows user to sign-out.- Basic react native page with native elements for buttons, sliders, checkboxes etc.- Can use some templates for settings (react-native-settings-list)
Models and Services	<ul style="list-style-type: none">- Need to develop models and services, internal data structures and design-patterns for robust and extensible product implementation.- Models for: device, device-details, device-data-stream, settings, profile etc.- Services for: gps, device data stream, auth service, rest apis service etc.
Testing and CI/CD Support	<ul style="list-style-type: none">- Complete Unit Test Cases, Integration Test Cases for each modules, model, services and screens.- Integrate into Jenkins/Travis based CI/CD framework for continuous integration, delivery and deployment.- Build iOS platform image on each check-in trigger.

Risk and Mitigation

Due to the tight deadline for this project - we have identified following as the risk items and some mitigations plans.

- Integration with device management APIs, device data stream APIs (websockets), user management APIs.
 - Client will assign technical POC to talk to for any questions with respect to integration with these APIs.
 - Client will provide demo for these APIs, Contractor can verify these APIs using Postman or other such tools.
 - Client will facilitate joint debugging sessions if there are API integration issues.
- iOS App Store Approval Process:
 - Client will initiate iOS App Store Approval Process immediately.
 - Client will provide feedback and answers as needed for the Application approval process.

Sprint Plan

We will be working on bi-weekly sprint cycles which allows for sufficient client feedback loop. For each Sprint, we will come up with *User Stories* that need to be completed for that Sprint. These User-stories can be short user-stories or long-epic stories. Each user story will have acceptance criteria for that Sprint, which allows us to track the user-story's progress during that Sprint.

Each sprint will have following meetings with the client:

- **Sprint planning meeting:** This would be a 1hr – 2hr meeting with the client to discuss all the user where we will discuss all the user-stories for the sprint and prioritize these user stories. The acceptance criteria for the Sprint will also be discussed during this meeting.
- **Sprint Demo Meeting:** This will be a 1hr meeting with the client where we will demonstrate all the sprint deliverables. During this meeting, we will demonstrate the completed user-stories, the testing done for those stories and how they meet the acceptance criteria set for the user stories.
- **Integration Meetings:** This will be a series of recurring meeting with the client to discuss updates to REST API and integrate them on client side work.

Project Milestones

Sprint	Features/Milestone
Sprint1	<ul style="list-style-type: none">- Template App- Mockups Integration- HTML Conversion- React Componentization- React Router/Navigation
Sprint2	<ul style="list-style-type: none">- Location Services, Maps Display- Redux State Management- Device API Integration- Integrate Charts and Graphs for Display
Sprint3	<ul style="list-style-type: none">- Device Data Stream API integration- User Login/Registration API integration- Review/Feedback Integration- Bug Fixes and Resolving Feedback issues

Acceptance Criteria:

For each user-story, we will have acceptance criteria that is mutually agreed upon with the client. The acceptance criteria can include one or more of the following:

- Demonstration of feature or functionality that needs to be implemented for the user-story.
- Code review for the feature or functionality for the user-story.
- Unit Tests and Integration Tests
- Visual Look and Feel for the story.

If, for any reason, an acceptance criterion is not met for a particular user-story, Contractor shall prioritize fixing that user-story to meet the acceptance criterion prior to working on other user-stories.

Project Handover:

When the project is in final sprint, Contractor would immediately start working on Project Handover to the client.

As part of the handover the following documentation and/or training would be provided:

1. Project Code Reviews: Complete project code reviews to the client team to help understand the code-flow and how to sustain it on a long-term basis.



2. Project Training Documentation: For each module of the project, we would create a technical documents and memos which should help client to understand the code changes delivered.
3. Code-walk-through Videos: Contractor would create screencasts of project explanation and walk through videos that would help client to guide through key modules of the project.
4. On-site training: Contractor would provide on-site trainings throughout the duration of the project to help with training and handover.
5. Project Codebase and Licenses handover: Contractor would hand-over any and all code base, test-cases for the project and any software licenses to the client during the project handover.

Project Support:

This initial estimate does not include Project Support. We can estimate and plan a SOW which will provide details of SLA or T&M based project support as well as SoW based plan for any subsequent features and enhancements.

Pricing Estimate:

<This is a sample estimate, please contact us to get correct price quote for your project.>

Description	Estimate Hours	Hourly	Total Cost
Dev1: Senior JavaScript Developer with React Experience	240	\$ 60	\$ 14,400
Dev 2: Senior Program Manager Architect	60	\$ 75	\$ 4,500
Total			\$ 18,900

Invoicing and Payment Terms:

<Include any invoicing and payment terms here>

Confirmation:

<Signature Blocks>