****

**User Requirements Specification (URS)**

ShopT

Version: 1.0

Author: Anh Nguyen

Created date: 12-10-2019

Project: ShopT – React Native code challenge

[I. Introduction 3](#_Toc24457244)

[1. Purpose of the document 3](#_Toc24457245)

[2. Modification history 3](#_Toc24457246)

[3. Relevant documents 3](#_Toc24457247)

[II. Overview 4](#_Toc24457248)

[1. Code structure 4](#_Toc24457249)

[2. Technical 5](#_Toc24457250)

[3. Requirement 6](#_Toc24457251)

# Introduction

## Purpose of the document

Documents from the developer’s perspective. Documentation, detailing the functions of shopT help developers build the application.

## Modification history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Modification date** | **Author** | **Content summary** |
| 1.0 | 12-10-2018 | Anh Nguyen | Describe code structure, tech and requirement. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Relevant documents

|  |  |  |
| --- | --- | --- |
| **Name** | **Created date** | **Author** |
| UI Guideline | 12-10-2019 | Anh Nguyen |
|  |  |  |
|  |  |  |

# Overview

## Code structure



Redux structure with redux-thunk middlewares:



* Action type:
  + Action type
  + Default state
* Actions:
  + Actions: Get data netword respository and dispatch type from ‘**action type**’ to ‘**reducers ‘**
* Assets
  + All resources for app: images, fonts…
* Global
  + All default data or functions like: colors, data, strings, utils…
* Network
  + Net
    - Network Utils: Configure for fetch data from network
    - Server path: All api link
  + Respository: All functions to work with data
* Reducers
  + Manage store, state
* View: All pages view of application



* + Base view: default function, render view…
  + Views
  + Route: all routing page, default props
* Widgets: All components can reuse like: popup, header, textfield…

## Technical

* React native toolkit: react-native-cli
  + Why react-native-cli:
    - You can add native modules
    - You can use libraries that use native code in Objective-C/Java
    - The application will be smaller than other cli like Expo
    - Easy to debugging
* CSS framework: React Native StyleSheet
* API Service: Use Local data instead of network data

## Requirement

* Use Asyncstorage technologies to store the stock level of products even when the app is closed and reopened.
* Use Asyncstorage technologies to store user information to pre-fill fields in the checkout page when the user returns.
* Low stock and out of stock flexible in each item data.s
* Write unit-tests for testing component views & logics code.
* Use Travis-ci tool for running automated tests & deployment