|  |
| --- |
| **Wulib-Application** **Product Requirements Document** |
| John Durand |

Table of Contents Introduction........................................................... 2

Vision Statment.................................................................................................. 3

Project Overview...................................................................................... 4

Project Overview cont.-.......................................................................... 5

Functional requirements ................................................................................................................ 6

nonfunctional requirements ................................................................................................................ 7

Diagram ....................................................................................................... 8

Diagram ................................................................ 9

Architecture and Software ................................................................................ 10

System interface ........................................................................ 11

user interface ........................................................................ 11

Question to cover..................................................... 12

Dependencies...................................................... 12

Question to cover..................................................... 12

Apple App Store..................................................... 13

Google Play ..................................................... 14

Technology Stack..................................................... 15

List of features..................................................... 16

List of features con-..................................................... 17

Requriements..................................................... 18

Requirement’s con-..................................................... 19

Passenger App..................................................... 20

Driver App..................................................... 21

Admin Panel..................................................... 22

Dispatch Panel..................................................... 22

Reference Page ............................................................ 23,24,25,26,27

**Vision Statement -**

**Mission Statement -**

Project overview document

[Woulib Project - see below]

[06/09/21]

Woulib Project

What will this project have?

Project overview statement

What will this project offer and how will it be different from others like it.

Introduction / Background to project

Business case

Deliverables- is a tangible or intangible good or service produced as a result of a project that is intended to be delivered to a customer. A deliverable could be a report, a document, a software product, a server upgrade or any other building block of an overall project

Constraints

Key people / Key stakeholders

Project lead or project manager

Client - or named contact in the area of the business the project is being carried out for

project administration

In scope

In depth on what will the project have

Out of scope

What the project won’t offer upon release

**Functional requirements** - defines the basic system behavior or what the system does or must do

* Descriptions of data to be entered into the system
* Descriptions of operations performed by each screen
* Descriptions of workflows performed by the system
* Descriptions of system reports or other outputs
* Who can one enter the data into the system?
* How the system meets applicable regulatory requirements

### **Non-Functional requirements** - Shows specify how the system should complete an induvial task

### Examples:

Capacity- the maximum amount that something can contain

Security - What are the security requirements, both for the physical installation and from a cyber perspective?

Usability- This focuses on the appearance of the user interface and how people interact with it. What color are the screens? How big are the buttons?

Maintainability- is defined as the probability of performing a successful repair action within a given time.

Serviceability- the quality of being able to provide good service. serviceableness, usability, useableness, useableness. usefulness, utility - the quality of being of practical use.

Reliability- the degree to which the result of a measurement, calculation, or specification can be depended on to be accurate.

Performance- How fast does it need to operate?

Data Integrity

Availability

Environmental

Data Integrity

Manageability

*Diagrams*

Workflow Diagram- A workflow diagram is a visual representation of a business process usually done through a flowchart. It uses standardized symbols to describe the exact steps needed to complete a process, as well as pointing out individuals responsible for each step.

Activate Diagram- describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.

USER CASE-describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.

Entry Relationship Diagram- describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.

System sequence Diagram- A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Unified Modeling Language Diagram- To help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems

Data Deployment Diagram- In the UML, deployment diagrams are used to visualize the static aspect of these physical nodes and their relationships and to specify their details for construction.

*Architecture and Software*

**Language used**

JavaScript- can be used to calculate supply and predict demands.

Nodes.js & Redis- Includes the real time dispatch system used

Java & objective-C is used for android and IOS apps.

Push Notification for apple implemented through App Push Notification Service and for google it’s Google Cloud Messaging

Python

SQLAlchemy

PHP

Ruby on Rails

Go

Mapkit

CoreLocation

Apple Push

Google Firebase

In-app payments gateway adapters (stripe, Braintree, Paytm, PayPal Mobile, Crypto, Apple pay, Google wallet) implementation.

Gradle

OkHttp

Retrofit

Gson for networking

MySQL

Monolithic software architecture model

DISCO

(APN) Apple Push Notification (FCM) Firebase Cloud Messaging- Twilio, Nexmo, Plivo, Sinch

System Interfaces- The interface between two (or more) systems describes how the systems “communicate” by allowing mass, energy, and information to flow between them.

user interface- (UI) is anything a user may interact with to use a digital product or service. This includes everything from screens and touchscreens, keyboards, sounds, and even lights. To understand the evolution of UI, however, it’s helpful to learn a bit more about its history and how it has evolved into best practices and a profession.

Questions to cover

* What platforms will the app you use (iOS, Android, or Windows, Google phone)?
* What operating system versions should support it?
* What are your current services, servers, databases?
* What are your maintenance needs? Do you need to support it for the future?
* How long should the app function before an overhaul is needed?
* Do you have current API/services documentation?
* Are there other credentials that are needed or already exist (analytics systems, or platforms)?

## Dependencies- Dependencies are any aspect that the product or product team relies on to meet objectives. These may include:

* Hardware that the app will run on/communicate with (for example, beacons)
* Service/API documentation
* Profile/account/platform credentials
* Any third-party software your app relies on
* Any flowcharts, documents, or information related to the product

## Constraints- Constraints are the limitations that teams must work within, typically related to scope, budget, and time. However, they may also include aspects like risk tolerance, resources/staff, and quality requirements.

## 

## Apple App Store

* iTunes Connect Account access
* Company/Entity Name
* App Store app listing name
* Search keywords
* Bundle id / SKU
* Demo account for reviewers
* Description
* Support URL
* Marketing URL
* Privacy policy
* App category
* Copyright information
* Contact information
* App icon (1024×1024)
* [App Store distribution provision profile](https://clearbridgemobile.com/how-to-create-a-distribution-provisioning-profile-for-ios/)
* App Store distribution code signing identity
* Screenshots (correct sizes based on devices)

## Google Play

* Google Play Developer access
* Store listing name
* Paid/free
* Short description
* Full description
* App icon (512×512)
* Feature Graphic (1024×500)
* App type
* App category
* Content Rating
* Contact Email
* Privacy Policy
* Screenshots (correct sizes based on devices)

Your mobile app requirements document should include all technical assets and information required for [Apple’s App Store submission](https://clearbridgemobile.com/how-to-submit-an-app-to-the-app-store/) and [Google Play submission](https://clearbridgemobile.com/how-to-submit-an-app-to-the-google-play-store/).  Defining these requirements in the early stages of a project will significantly expedite the submission process when the product is ready for release. While these will vary depending on the app stores being submitted to, below are the assets and information to include for the Apple App Store and Google Play.

Technology stack-

identifying a device’s location

Providing driving directions

Integrating with mapping software

Push Notification and SMS

Payment integration

Price calculator

Booking feature for drivers

Ability to identify device location

Point to point direction

The ability to receive alerts

The ability to see the driver’s profile and status

The ability to see the route from Their Phones

Requesting previous drivers

Waitlist instead of surge pricing

Pickup Location

Split payment

Interactive Map

Later ride

Book for others

Voice recognition

Waiting list

Panic Button

Trip Alert

Driver report

Destination

Forward Dispatch

Cost per mile

Cost per minute

Base fare

Booking fee

## List Of Features

* Sign-up and login
* Onboarding
* Navigation
* Social media integration
* Social Feeds
* Booking systems
* Calendar integrations
* Push notifications
* Native maps
* Device hardware access
* App analytics
* Geolocation
* Push Notification
* Payments
* Register
* Messages
* Price Calculator
* Rating& reviews
* Booking history
* Report
* Driver interface
* Navigation
* Price estimation
* SMS send out
* Log in with social media
* Touch ID/Face ID login
* Saved Password support
* User walkthrough
* Destination picker
* Rideshare
* Taxi booking
* Driver rates & reviews
* Customer support
* Trip Cancellation fee
* Live location (of driver)
* Pick-up location
* Drop-off location

## Requirements

* The following considerations to include when mapping out business requirements:
* What is the purpose of the app or product? What are you trying to accomplish?
* What is the current problem(s) it will solve?
* How will it improve the current process? Will it facilitate a new process?
* Will the app need to be started from scratch, or can you leverage existing assets?
* What should the app be able to do? What is the product’s core functionality?
* What features will it need?
* Are there branding and design guidelines to follow?
* Locations & Fares Management
* Driver & User Management
* Booking Management
* Vehicle Management
* Review Driver’s Orders & Payoffs
* Check Ratings & Reviews
* Notifications Management
* System Content Management & FAQ
* Promotions & Discounts Management
* Driver & User Support
* Integration with Google Analytics
* Driver app
* Passenger app
* Dispatch panel
* Admin panel

Passenger app

* Register/log-in
* Profile creation
* Push notifications
* Payments
* Fare calculator
* Ride history
* In-built messaging
* Driver rating & review
* Vehicle selection
* Split payment
* Tracking taxi
* Interactive map

Driver app

* Registration
* Profile creation
* Accept/decline the ride request
* Passenger ratings & reviews
* Push notifications
* Reports
* Pick-up & drop-off navigation
* Trip history
* Cost estimation

Admin panel

* Driver management
* Vehicle management
* Payment management
* Ratings & reviews management
* Complaint management
* Earnings report
* Performance report

Dispatch panel

* Fleet management
* Tracking trips
* Manage service requests
* Matching drivers & passengers
* Reviews & feedback management

Reference page

<https://deseng.ryerson.ca/dokuwiki/design:system_interface>

<https://clearbridgemobile.com/how-to-build-a-mobile-app-requirements-document/>

<https://yalantis.com/blog/uber-underlying-technologies-and-how-it-actually-works/>

<https://medium.com/nerd-for-tech/uber-architecture-and-system-design-e8ac26690dfc>

<https://www.guru99.com/non-functional-requirement-type-example.html>

Diagram

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram

Description automatically generated

Table

Description automatically generated

