Software Requirements Specification

for

Q-Taskers

**Version 2.0**

**Prepared by Karan Inder Singh**

**Q-Taskers**

**24/09/2018**

**Table of Contents**

**Table of Contents [ii](#_25b2l0r)**

**Revision History [ii](#_30j0zll)**

**1. Introduction [1](#_kgcv8k)**

1.1 Purpose [1](#_34g0dwd)

1.2 Document Conventions [1](#_1jlao46)

1.3 Intended Audience and Reading Suggestions [1](#_43ky6rz)

1.4 Product Scope [1](#_2iq8gzs)

1.5 References [1](#_xvir7l)

**2. Overall Description [2](#_3hv69ve)**

2.1 Product Perspective [2](#_1x0gk37)

2.2 Product Functions [2](#_4h042r0)

2.3 User Classes and Characteristics [2](#_2w5ecyt)

2.4 Operating Environment [2](#_1baon6m)

2.5 Design and Implementation Constraints [2](#_3vac5uf)

2.6 User Documentation [2](#_2afmg28)

2.7 Assumptions and Dependencies [3](#_pkwqa1)

**3. External Interface Requirements [3](#_39kk8xu)**

3.1 User Interfaces [3](#_ihv636)

3.2 Hardware Interfaces [3](#_1opuj5n)

3.3 Software Interfaces [3](#_48pi1tg)

3.4 Communications Interfaces [3](#_2nusc19)

**4. System Features/Functional Requirements [4](#_1302m92)**

4.1 System Feature 1 [4](#_3mzq4wv)

4.2 System Feature 2 (and so on) 4

**5. Prioritization and Release Plan**

5.1 Choice of prioritization method 4

**6. Other Nonfunctional Requirements [4](#_haapch)**

6.1 Performance Requirements 4

### 7.0 Safety Requirements

7.1 Security Requirements

7.2 Software Quality/System Attributes

7.3 Security Requirements

7.4 Business Rules

**8. Other Requirements [5](#_upglbi)**

**Appendix A: Glossary [5](#_3ep43zb)**

**Appendix B: Analysis Models [5](#_1tuee74)**

**Appendix C: To Be Determined List**

**9. Use Case Diagram**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Revision 2.0 | 24/9/18 | Updating and finalising the document | 2.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of the document is to represent an online system for offering services like Electronics and Electrical servicing, Tutors, Photography, Planning, etc. (logistics for households) on a single platform and hence removing the middlemen who are involved.

## Document Conventions

1. Default sentences/ Body of Document (Times 12).

## Bold words are headings and subheadings (Times 14+ Bold).

## Intended Audience and Reading Suggestions

This project is a prototype of an online service-offering platform which targets every section of the society who are willing to enjoy a single platform for any paid-service they are supposed to get. This is also intended for the shops and individuals providing the services.

## Product Scope

The purpose of this all-inclusive platform is to offer a plethora of ultra-fast services. The customers don’t have to contact and search for each of them at different places and still wait to get them. This also would help the shops to get a regular customer base regardless of weather, time of the year and therefore biased-brokerage is eliminated.

## References

1. <http://qtaskers.herokuapp.com/>
2. <https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc>
3. <http://searchsoftwarequality.techtarget.com/answer/Software-requirements-specification-and-the-IEEE-standard>

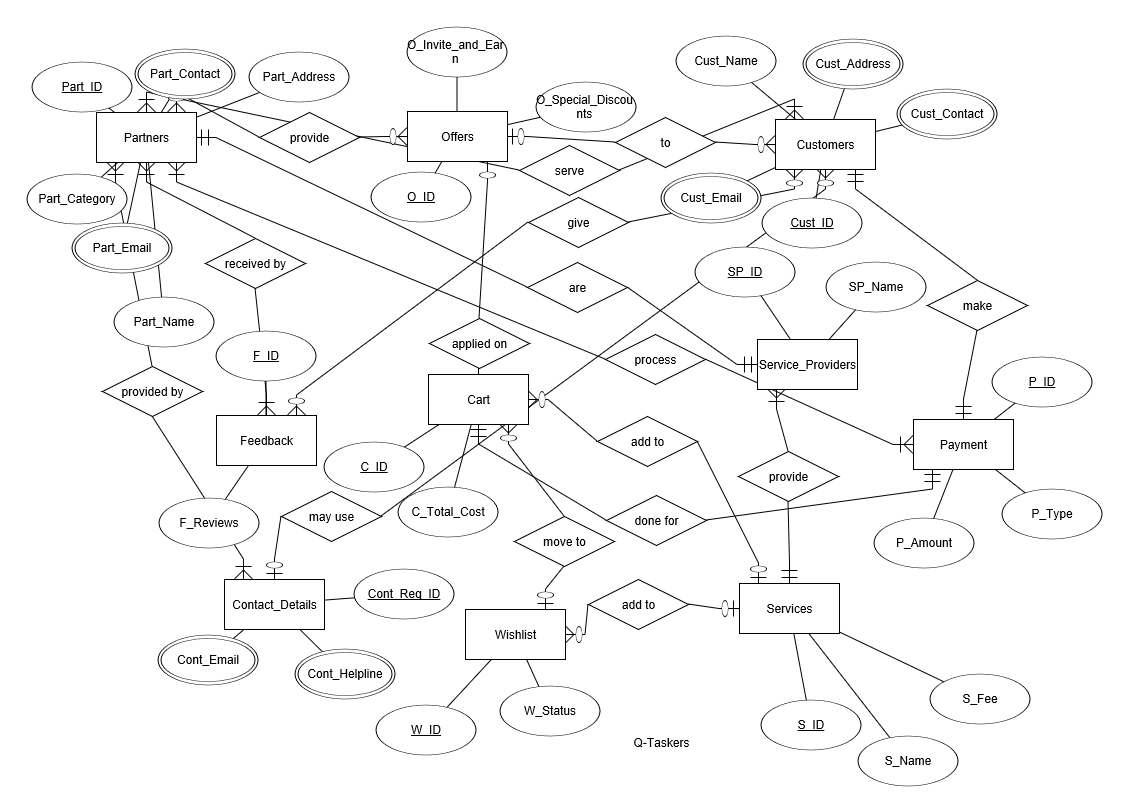
# Overall Description

## Product Perspective

This is an online platform offering services and is an improvement over previously present sites like UrbanClap and TaskRabbit with a different business model altogether and also a different market penetration level and target.

## Product Functions

The product functions have been explained using the following ER diagram

**

## User Classes and Characteristics

1. Business Partners: Information about those shops who register with the app.
2. Service Providers: Details about the people who are to actually offer the services at the doorstep.
3. Customers: The individuals who are to register with us and enjoy the doorstep services offered.
4. Guests: The individuals who are not registered but still want to enjoy the doorstep services offered.

Registration is done using name, mobile no. , address, email (optional).

Login will be done on a standard login portal which uses email/phone no. and password for authentication. Login portal will have a sign up as Guest option for Guest users.

The app acts as a bridge between all these people to have a win-win situation for everyone.

## Operating Environment

Operating environment for the services-offering system is as listed below.

1. OS: Windows 7 sp1 and later(amd64), Android(4.0 and later versions) (arm64)
2. Database(Distributed Database): SQLite/MongoDB/NoSQL
3. Platforms: Java-JavaFX/PHP/HTML

## Design and Implementation Constraints

1. The global schema and the allocation schema
2. SQL commands for the queries
3. A category-based interface to access the services
4. Verification of the customers using OTP-authentication and services can be offered or enjoyed by verified personnel only.

## User Documentation

User manual for Q-Taskers windows app

Help and video tutorials in the android app

SRS IEEE document for Q-taskers

## Assumptions and Dependencies

The closest verified service provider will be allocated to the user using Google Maps API and it is assumed that once a user requests a second service, the same provider will be allocated to him to increase the loyalty benefits for him.

# External Interface Requirements

## User & Software Interfaces

Simple UI/UX using Studio by UXPin

Multilingual

Buttons- Settings, Help, About

Help will be interactive which will be based on text messages which will be displayed along with a tutorial.

Additional tutorials for usage and settings change will be added on the website and within the app as well.

App will have a link to the product documentation in the ‘About’ section.

The About section will include additional details like built using, version, build no., which will be helpful to the developers.

## Hardware Interfaces

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

## Communications Interfaces

The mobile application communicates with the GPS application in order to get geographical information about where the user is located and the visual representation of it, and with the database in order to get the information about the service providers. The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while the communication between the database and the mobile application consists of only reading operations. Email of customer may be used to get review/satisfaction of service most recently used. SHA hashing mechanism will be used for password hashing. 1 Mb/s is the minimum data rate for efficient usage & functioning of the app. HTTP/HTTPS will be used between client and server.

# System Features/ Functional Requirements

### 4.1 User Class 1 - The User

***4.1.1 Functional requirement 1.1***

#### ID: FR1

TITLE: Download mobile application

DESC: A user should be able to download the mobile application through either an application store or similar service on the mobile phone. The application should be free to download. Reason: In order for a user to download the mobile application. Dependency: None

***4.1.2 Functional requirement 1.2***

#### ID: FR2

TITLE: Download and notify users of new releases

DESC: When a new/updated version or release of the software is released, the user should check for these manually. The download of the new release should be done through the mobile phone in the same way as downloading the mobile application.

Reason: In order for a user to download a new/updated release.

Dependency: FR1

***4.1.3 Functional requirement 1.3***

#### ID: FR3

TITLE: User registration - Mobile application

DESC: Given that a user has downloaded the mobile application, then the user should be able to register through the mobile application. The user must provide user-name, password and e-mail address. The user can choose to provide a regularly used phone number. Reason: In order for a user to register on the mobile application. Dependency: FR1

***4.1.4 Functional requirement 1.4***

#### ID: FR4

TITLE: User log-in - Mobile application

DESC: Given that a user has registered, then the user should be able to log in to the mobile application. The log-in information will be stored on the phone and in the future the user should be logged in automatically.

Reason: In order for a user to register on the mobile application. Dependency: FR1, FR3

***4.1.5 Functional requirement 1.5***

#### ID: FR5

TITLE: Retrieve password

DESC: Given that a user has registered, then the user should be able to retrieve his/her password by email.

Reason: In order for a user to retrieve his/her password. Dependency: FR1

***4.1.6 Functional requirement 1.6***

#### ID: FR6

TITLE: Mobile application - Search

DESC: Given that a user is logged in to the mobile application, then the first page that is shown should be the search page. The user should be able to search for services, according to several search options. The search options are Price, Destination, Service type. There should also be a freetext search option. A user should be able to select multiple search options in one search. Reason: In order for a user to search for a service provider. Dependency: FR4

***4.1.7 Functional requirement 1.7***

#### ID: FR7

TITLE: Mobile application - Search result in a map view DESC:

• Search results can be viewed on a map. On the map, the relevant and closest services according to the user’s position are shown.

• A specific pin will represent a specific service provider location. On each pin there should be an information link.

• There should be maximally 100 results displayed. The map view should have a default zoom.

• The map view should include a button that, when selected, should display different filtering options in a filtering menu.

Reason: The way results are displayed in a map. Dependency:FR6

***4.1.8 Functional requirement 1.8***

#### ID: FR8

TITLE: Mobile application - Search result in a list view DESC:

• Search results can be viewed in a list. Each element in the list represents a specific service provider. Each element should include the service name, telephone number, distance according to the user’s position, average price, a link to the service’s web-page and an information link.

• There should be maximally 100 results displayed. If the result contains more services than what can be displayed on the screen at one time, the user should be able to scroll through them.

• When searching by price the services should be sorted according to the following order:

1. average price

2. distance

3. service type

• When searching by a search option, other than price, the service should be sorted according to the following order:

1. distance

2. average price

3. service type

• The list view should include a header with different selectable sorting options.

• The list view should include a button that, when selected, should display different filtering options in a filtering menu. Reason: The way results should be displayed in a list.

Dependency: FR6

***4.1.9 Functional requirement 1.9***

#### ID: FR9

TITLE: Mobile application - Switch result view

DESC: A user should be able to switch between a map view and a list view for all search options.

Reason: In order for a user to switch between result views.

Dependency FR7, FR8

***4.1.10 Functional requirement 1.10***

#### ID: FR10

TITLE: Mobile application - Search by price

DESC: A user should be able to input a maximum and a minimum price range. The result is displayed in a list view by default.

Reason: In order for a user to search by price. Dependency: FR8

***4.1.11 Functional requirement 1.11***

#### ID: FR11

TITLE: Mobile application - Search by service type

DESC: A user should be able to select a service type in a given list as input. The result is displayed in a map view by default.

Reason: In order for a user to search by service type.

Dependency: FR7

### 4.2 User Class 2 – Service Provider

***4.2.1 Functional requirement 2.1***

#### ID: FR22 Feature: Create an account

In order to create an account

A Service Provider

Should register on the web-portal

#### Scenario: Required information for registration

Given the service provider wants to create an account

And the service provider does not have an account

When the service provider registers on the web-portal by providing user-name

And password

And address

And e-mail address

And phone number

Then the service provider should be able to apply for verification

#### Scenario: Full information for registration

Given the service provider wants to create an account

And the service provider does not have an account

When the service provider registers on the web-portal by providing username

And password

And address

And e-mail address And phone number

And mobile number

Then the service provider should be able to apply for verification

#### Scenario: Confirmed registration

Given the service provider has applied for verification

And has not received a confirmation e-mail after registration

When the service provider receives a confirmation e-mail

Then the service provider should be able to log in

***4.2.2 Functional requirement*** 2.2

#### ID: FR23 Feature: Service provider log-in

In order to use the system

A service provider

Should be logged in to the web-portal

#### Scenario: Successful log-in

Given the service provider wants to log in

When the service provider logs in with his/her account

Then the service provider should be logged in as a service provider

#### Scenario: Retrieve password

Given the service provider wants to log in

And has lost the password

When the service provider enters his/her email address in the “Retrieve password” form

And submits the form

Then the service provider should receive an email containing the password

***4.2.3 Functional requirement 2.3***

#### ID: FR24 Feature: Manage information

In order to manage information

A service provider

Should be logged in to the web-portal

#### Scenario: Show fields for managing information

Given the service provider is logged in

When the service provider wants to manage information

Then the service provider should be able to manage information in a form

**Scenario: Filling in mandatory fields**

Given the service provider wants to fill in the mandatory fields of the form

When the service provider provides average price

And address

And e-mail address

And phone number

And service provider name

Then the service provider has filled the mandatory fields of the form

#### Scenario: Filling in optional fields

Given the service provider of a service provider wants to fill in optional fields in the form

When the service provider provides service provider description

And menu

And type of service provider

And picture of service provider

And mobile phone

Then the service provider has filled in optional fields in the form

#### Scenario: Adding information with mandatory fields

Given the service provider has filled in the mandatory fields of the form

When the service provider submits the form

Then the information about the service provider should be added

#### Scenario: Adding information with mandatory and optional fields

Given the service provider has filled in the mandatory fields of the form

And filled in one or more optional fields of the form

When the service provider submits the form

Then the information about the service provider should be added

#### Scenario: Deleting information

Given the service provider is logged in

And information exists

When the service provider deletes information

Then the information should be deleted

#### Scenario: Editing information

Given the service provider is logged in

And information exists

When the service provider edits information

Then the information should be edited

### 4.3 User Class 3 - Administrator

***4.3.1 Functional requirement 3.1***

#### ID: FR26 Feature: Administrator log in

In order to administer the system

An administrator

Should be logged in to the web-portal

#### Scenario: Successful log-in

Given the administrator wants to log in

When the administrator logs in with an administrator account

Then the administrator should be logged in as an administrator

***4.4 Functional requirement 4***

**ID: FR27**

#### Feature: Verify service provider

In order to allow a service provider to use the system

An administrator

Should be able to verify the service provider

#### Scenario: Verify a service provider

Given the administrator is logged in

When the administrator verifies a service provider

Then the service provider should be able to log in

And the service provider should be notified by a confirmation email

#### Scenario: Reject a service provider

Given the administrator is logged in

When the administrator rejects a service provider

Then the service provider should not be able to log in

And the service provider should be notified by a rejection email

***4.3.3 Functional requirement 3.3***

#### ID: FR28 Feature: Manage service provider types

In order to have a list of service provider types

An administrator

Should be able to manage the service provider types

#### Scenario: Add a new service provider type

Given the administrator is logged in

When the administrator creates a new service provider type

Then the new service provider type should be added to the list of service provider types

**Scenario: Editing an existing service provider type**

Given the administrator is logged in

When the administrator edits an existing service provider type

Then the service provider type should be updated in the list of service provider types

#### Scenario: Delete a service provider type

Given the administrator is logged in

When the administrator deletes a service provider type

Then the deleted service provider type should be removed from the list of service provider types

***4.3.4 Functional requirement 3.4***

#### ID: FR29 Feature: Manage service provider services

In order to have a list of services

An administrator

Should be able to manage the services

#### Scenario: Add a new service

Given the administrator is logged in

When the administrator creates a new service

Then the new service should be added to the list of services

**Scenario: Editing an existing service** Given the administrator is logged in

When the administrator edits an existing service

Then the service should be updated in the list of services

#### Scenario: Delete a service

Given the administrator is logged in

When the administrator deletes a service

Then the deleted service should be removed from the list of services

***4.3.5 Functional requirement 3.5***

#### ID: FR30 Feature: Manage service provider information

In order to manage service provider information

An administrator

Should be logged in to the web-portal

#### Scenario: Add service provider information

Given the administrator is logged in

When the administrator adds service provider information

Then the information should be added to the service provider

#### Scenario: Delete service provider information

Given the administrator is logged in

And information about a service provider exists When the administrator deletes the information

Then the information about the service provider should be deleted

#### Scenario: Edit service provider information

Given the administrator is logged in

And information about a service provider exists

When the administrator edits the information

Then the information about the service provider should be edited

***4.3.6 Functional requirement 3.6***

#### ID: FR31 Feature: Manage users

In order to keep track of the users

An administrator

Should be able to manage the users

#### Scenario: Edit an existing user’s information

#### Given the administrator is logged in

When the administrator edits an existing user

Then the user information should be updated

#### Scenario: Delete/Inactivate an existing user

Given the administrator is logged in

When the administrator deletes an existing user

Then the user should be deleted

***4.3.7 Functional requirement 3.7***

#### ID: FR32 Feature: Manage service providers

In order to keep track of the service providers

An administrator

Should be able to manage the service providers **Scenario: Add a new service provider**

Given the administrator is logged in

When the administrator creates a new service provider Then the new service provider should be added

#### Scenario: Edit an existing service provider

Given the administrator is logged in

When the administrator edits an existing service provider Then the service provider information should be updated

#### Scenario: Delete an existing service provider

Given the administrator is logged in

When the administrator deletes an existing service provider

Then the service provider should be deleted

And the service provider information should be deleted

# 5. Prioritization and Release Plan

In order to get a view of how to divide the requirements into different releases and what requirements should be included in which release, a prioritization of the requirements is needed. This section discusses the choice of prioritization methods and gives a suggestion of how the release plan for these requirements could look like.

## 5.1 Choice of prioritization method

When prioritizing the requirements the ten most important ones were picked out first. This was done with a simple “1 to 10” ranking method, with one being “not important” and ten “very important”. Based on the elicitation meetings, and the perceived ideas of what was important to the different stakeholders, a number was set for each requirement. The numbers were then summed up for each requirement and the ten with the highest score were chosen to be prioritized with the cost value approach.

**5.1.1 Description and Priority**

Use of maps (Google Maps) for location based suggestions and implementation of the app.

Integration of Rating based services and suggestions for implementation of app.

**5.1.2 Stimulus/Response Sequences**

*Not Applicable*

**5.1.3 Functional Requirements**

*mentioned above in Section 4.*

# 6. Other Nonfunctional Requirements

## 6.1 Performance Requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

***6.1.1 Prominent search feature***

#### ID: QR1

TITLE: Prominent search feature

DESC: The search feature should be prominent and easy to find for the user.

RAT: In order to for a user to find the search feature easily.

DEP: none

***6.1.2 Usage of the search feature***

#### ID: QR2

TITLE: Usage of the search feature

DESC: The different search options should be evident, simple and easy to understand.

RAT: In order to for a user to perform a search easily.

DEP: none

***6.1.3 Usage of the result in the list view***

#### ID: QR3

TITLE: Usage of the result in the list view

DESC: The results displayed in the list view should be user friendly and easy to understand. Selecting an element in the result list should only take one click. RAT: In order to for a user to use the list view easily. DEP: none

***6.1.4 Usage of the result in the map view***

#### ID: QR4

TITLE: Usage of the result in the map view

DESC: The results displayed in the map view should be user friendly and easy to understand. Selecting a pin on the map should only take one click.

RAT: In order to for a user to use the map view easily.

DEP: none

***6.1.5 Usage of the information link***

#### ID: QR5

TITLE: Usage of the information link

DESC: The information link should be prominent and it should be evident that it is a usable link.

Selecting the information link should only take one click.

RAT: In order to for a user to use the information link easily.

DEP: none

***6.1.6 Response time***

#### ID: QR6

TAG: Response Time

GIST: The fastness of the search

SCALE: The response time of a search

METER: Measurements obtained from 1000 searches during testing.

MUST: No more than 2 seconds 100% of the time.

WISH: No more than 1 second 100% of the time.

***6.1.7 System dependability***

#### ID: QR7

TAG: System Dependability

GIST: The fault tolerance of the system.

SCALE: If the system loses the connection to the Internet or to the GPS device or the system gets some strange input, the user should be informed.

METER: Measurements obtained from 1000 hours of usage during testing. MUST: 100% of the time.

# 7. Safety Requirements

*Deleting of the data folder and/ or AppData of the app may result in loss of user data.*

## 7.1 Security Requirements

*No other personnel to authorised to tamper with the source code and database of the app.*

## 7.2 Software Quality/System Attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

***7.1 Reliability***

#### ID: QR8

TAG: SystemReliability

GIST: The reliability of the system.

SCALE: The reliability that the system gives the right result on a search.

METER: Measurements obtained from 1000 searches during testing.

MUST: More than 98% of the searches.

PLAN: More than 99% of the searches.WISH: 100% of the searches.

***7.2 Availability***

#### ID: QR9

TAG: SystemAvailability

GIST: The availability of the system when it is used.

SCALE: The average system availability (not considering network failing).

METER: Measurements obtained from 1000 hours of usage during testing.

MUST: More than 98% of the time.

PLAN: More than 99% of the time. WISH: 100% of the time.

#### ID: Q10

TITLE: Internet Connection

DESC: The application should be connected to the Internet.

RAT: In order for the application to communicate with the database. DEP: none

**ID: QR11**

TITLE: GPS Connection

DESC: The application should be connected to the GPS device.

RAT: In order for the application to get the users location, the map and to calculate the distance.

DEP: none

***7.3 Security***

#### ID: QR12

TAG: CommunicationSecurity

GIST: Security of the communication between the system and server.

SCALE: The messages should be encrypted for log-in communications, so others cannot get user-name and password from those messages.

METER: Attempts to get user-name and password through obtained messages on 1000 log-in session during testing.

MUST: 100% of the Communication Messages in the communication of a log-in session should be encrypted.

Communication Messages: Defined: Every exchanged of information between client and server.

## 7.3 Business Rules

*Only verified and qualified technicians would be hired to incorporate with us.*

# 8. Other Requirements

Not Applicable

**Appendix A: Glossary**

RAT:Rational/Reason

**Appendix B: Analysis Models**

TBD

**Appendix C: To Be Determined List**

TBD

# 9. Use Case Diagram

