Software Requirements Specification

for

Q-Taskers

Version 3.0

Prepared by Karan Inder Singh

Q-Taskers

10/10/2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Revision 3.0	10/10/18	Elaborated Below	3.0

Reason for changes:

- Use case diagram
    - Add admin user and its implementation
- Index change according to original
- Topics according to original
- Add Product function
- Diagrams to be placed in the end
- Add Operating Environment
- User classes and characteristics

1. Introduction

1.1 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.

1.2 Document Conventions

1. Default sentences/ Body of Document (Times 12).
2. **Sub-Headings (Times 14).**
3. **Headings (Times 18)**

1.3 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.

1.4 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.

1.5 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

2. Overall Description

2.1 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. This system will consist of two parts: one mobile application and one web portal, one each for the customer and the service provider.. For the customer, the mobile application will need to communicate to a GPS application, Google Maps within the mobile phone, which in turn communicates with a physical GPS device to find the location of the user,. The GPS will provide the mobile application with locations of both the user and the service provider and the distance between them. The same for the web-portal will be provided by the location services. Since this is a data-centric product, it will need somewhere to store the data of those who register and request a service. For that, a database will be used. For the service providers, on either the web-portal or the app, registration forms and verification parameters are there, successfully filling those will store the data in a database.

2.2 Product Functions

Client Side:
The users will be able to register online or login with a login name and password or they might also go unregistered. Then he/she can browse or search the multiple services offered. Selecting the service will direct the registered user to view the feasibility of the service in the area and its estimated cost and then the option for payment, i.e., the COD will be displayed. The same for a guest(unregistered) user will ask him/her to add his details and also an option to sign-up, doing the same will show him the estimated details Reaching the payment page and confirming the COD will show the contact of the service provider to which he/she may immediately. communicate else anyway, the serviceman will visit his/her place and the requested service will be provided.

Service Provider Side:
The partners will be able register with us and will be needed to confirm to some verification parameters. Initially, the service provider will be shared the requests of the customers without any charge, and no profit-sharing would be asked till the period of establishing the business. After the business flourishes, the business model would be converted to a B2C model and the providers will be charged a fixed cost for a fixed time and then they would be allocated with the customers. The details of the nearest customer and the request will be shared with them and they would be asked to confirm to the request. They would be required to complete this request ASAP.

2.3 User Classes and Characteristics

1. Admin: It includes us as a middle-men who work between the customers and the service providers helping them connect via the platform and check the flow of work. They have analytics and system change rights.

2. Service Providers: Details about the people who are to actually offer the services at the doorstep. Information about those shops who register with the already hosted website (qtaskers.herokuapp.com).

3. Registered users: The individuals who are to register with us and enjoy the doorstep services offered.

4. Guest users: The individuals who are not registered initially. They surf for services, register themselves and then they are ready 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.

2.4 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

2.5 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.

2.6 User Documentation

User manual for Q-Taskers windows app

Help and video tutorials in the android app
SRS IEEE document for Q-taskers

2.7 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.

3. External Interface Requirements

3.1 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.

3.2 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.

3.3 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.

4. 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 username, password and email 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 login - 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 email after registration
When the service provider receives a confirmation email
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.

6. System Features

6.1 System Feature 1~Google Maps API

6.1.1 Description and Priority

High priority feature. Calculation of distance between service provider and user to provide cost efficient and fast services at user's doorstep.

6.1.2 Stimulus/Response Sequences

On using the Google Maps API, the current location of the user would be retrieved and using the pincode, the feasibility of the service offered and the estimated cost will be calculated.

6.1.3 Functional Requirements

mentioned in section 4

6.2 System Feature 2 ~Ratings and Feedback

6.2.1 Description and Priority

Medium priority feature. The Performance of the service providers is reviewed and then the chances for their re-allotment to any future request of the same customer will be decided.

6.2.2 Stimulus/Response Sequences

On using the feature, the customer feedback will be noted and stored for future references

6.2.3 Functional Requirements

mentioned in section 4

7. Other Nonfunctional Requirements

7.1 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.

7.1.1 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

7.1.2 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

7.1.3 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 listview easily. DEP: none

7.1.4 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

7.1.5 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

7.1.6 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.

7.1.7 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.

7.2. Safety Requirements

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

7.3. Security Requirements

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

7.4. Software Quality/System Attributes

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

7.4.1 Reliability

ID: QR8
TAG: System Reliability

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.

7.4.2 Availability

ID: QR9
TAG: System Availability
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

7.4.3 Security

ID: QR12
TAG: Communication Security
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.

7.5 Business Rules

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

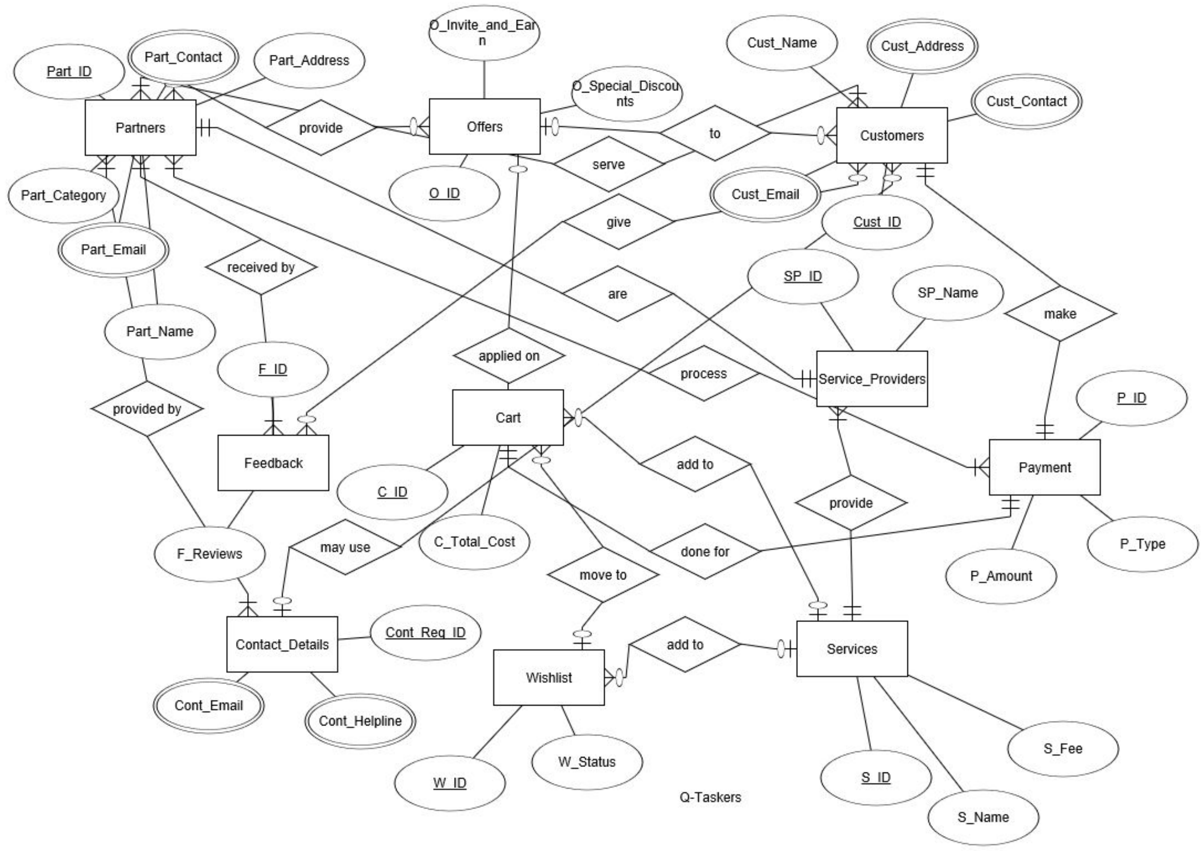
8. Other Requirements

Not Applicable

Appendix A: Glossary

RAT:Rational/Reason

Appendix B: Analysis Models

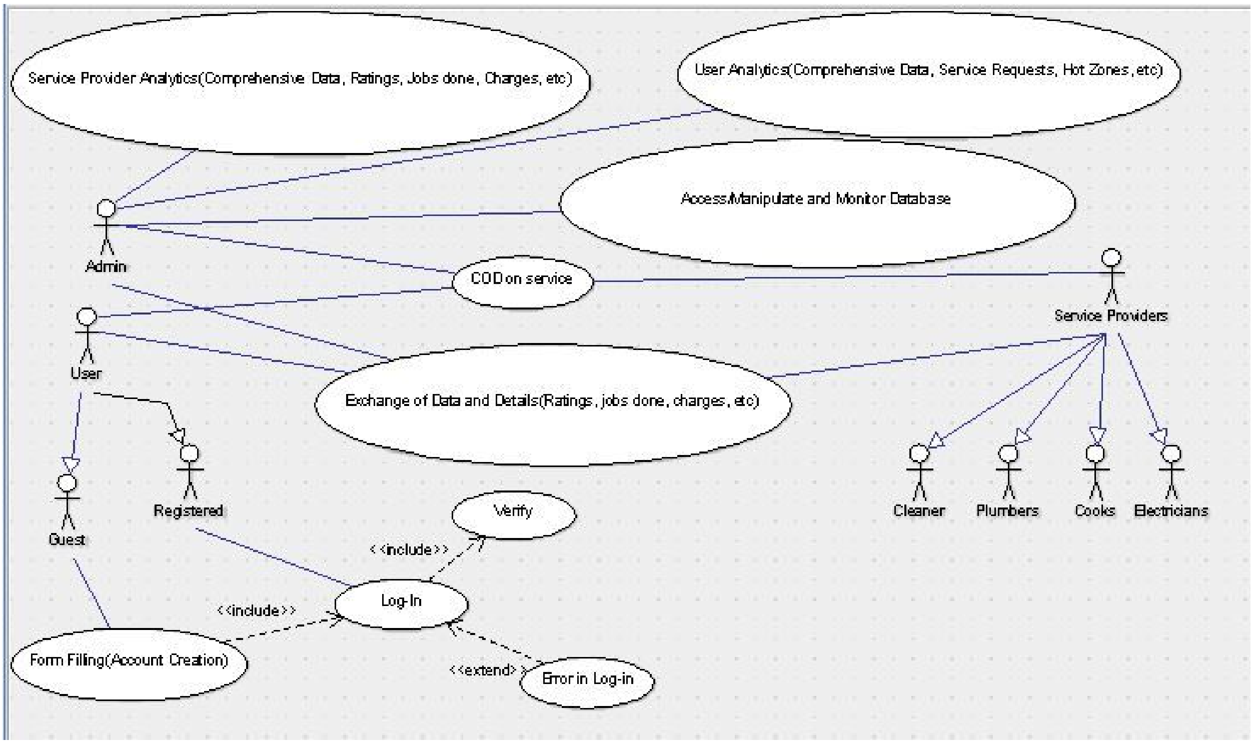


TBD

Appendix C: To Be Determined List

TBD

Appendix D: Use Case Diagram


Service Provider Analytics(Comprehensive Data, Ratings, Jobs done, Charges, etc)
User Analytics(Comprehensive Data, Service Requests, Hot Zones, etc)
Access/Manipulate and Monitor Database
Admin
COD on service
Service Providers
User
Exchange of Data and Details(Ratings, jobs done, charges, etc)
Guest
Registered
Cleaner
Plumbers
Cooks
Electricians
Verify
<<include>>
Log-In
<<include>>
Form Filing(Account Creation)
<<extend>>
Error in Log-in