1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete HandyMan (online home servicing system) by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the HandyMan are provided in this document.

1.1 Purpose

The purpose of the project **HandyMan** is to provide service man for every service people need. People can find service provider for any problem and any task as well as the service providers also find task as bonus from online.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2 Project Scope

Primarily, the main scope of this project is to develop an application based on smartphone device. As more than 80% users of smartphones are using Android operating based mobile device, so we are targeting to implement our system firstly for Android users. Our proposed system has 2 modules.

This SRS is also aimed at specifying requirements of software to be developed but it can also be applied to assist in the selection of in-house and commercial software products. The standard can be used to create software requirements specifications directly or can be used as a model for defining a organization or project specific standard. It does not identify any specific method, nomenclature or tool for preparing an SRS.

1.3 Definitions Acronyms and Abbreviations

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document. Terms and concepts from the application domain are defined.

Service Provider	Who will receive work request via HandyMan and will provide service
Service Taker	Who will send work request via HandyMan, hire the service provider and will receive service
FAQ	Frequently Asked Questions

1.4 References

Software & Systems Requirements Engineering: In Practice forewards by Manfred Broy, technical University, Munich, and Erik Simmons, Intel Corporation

1.5 Overview

Day by day people's business is increasing. Besides if our daily-used things like refrigerator, air condition, gas stove and other things get wasted or any other problem occurs like those then it becomes quite irritating and hard to carryout those somewhere to repair or to find correct mechanic to repair. To remove this burden our online platform is here to provide quality service providers for almost everything.

It will not only provide service providers to people, but also provide job to the service providers.

Now-a-days maximum of our population use smartphone and 90% of them are based on Android operating system. So that our project HandyMan will be an android application to reach people widely and easily.

2. User classes and characteristics

There are 3 types of stakeholders in our "HandyMan" app. Such as:

Service Provider: A service provider is a vendor that provides IT solutions and/or services to end users and organizations. This broad term incorporates all IT businesses that provide products and solutions through services that are on-demand, pay per use or a hybrid delivery model. A service provider's delivery model generally differs from conventional IT product manufacturers or developers. Typically, a service provider does not require purchase of an IT product by a user or organization. Rather, a service provider builds, operates and manages these IT products, which are bundled and delivered as a service/solution. In turn, a customer accesses this type of solution from a service provider via several different sourcing models, such as a monthly or annual subscription fee.

Service Taker: Services have subscription payment models rather than point-of-sale models. For example. HandyMan is a service provider, not movie retailer. Software as a service is a more lucrative model than selling a single unlimited license. For example, Microsoft makes less money selling you a copy of Office than selling you a subscription to Office 365, which is a service. The benefits to the consumer include continuous improvement of the software over time. Google offers App Engine which is a platform as a service. The platform can run your service with features like scaling to support any load of traffic. In this model they are selling access to their infrastructure as a service. That kind of service cannot easily be packaged and sold as a product. They tried selling hardware at one point to encapsulate their search technology as a

physical product. I don't think it saw widespread adoption and they stopped doing that. I think they learned that services are easier to maintain and make more money.

bKash/DBBL Organization: bKash is a mobile financial service in Bangladesh operating under the authority of Bangladesh Bank as a subsidiary of BRAC Bank Limited. This mobile money system started as a joint venture between BRAC Bank Limited, Bangladesh and Money in Motion LLC, United States of America. bKash users can deposit money into their mobile accounts and then access a range of services, in particular transferring and receiving money domestically, making payments and can recharge prepaid mobiles easily. We use bKash in our app and also added DBBL

3. Design and Implementation Constraints

3.1 Interfaces

There are many types of interfaces as such supported by the HandyMan application system namely; User Interface, Software Interface and Hardware Interface.

The protocol used shall be HTTP.

The Port number used will be 80.

There shall be logical address of the system in IPv4 format.

3.1.1 User Interfaces

The user interface for the software shall be compatible to any android version higher than android 4.0 such as Android 4.4 Kitkat, Android 5 Lollipop, Android 6 Marshmallow, Android 7 Nogut, Android 8 Oreo, Android 9 Pie and the latest Android by which user can access to the system.

Also the user interface shall be implemented using any customized UI based on android 4+ .

3.1.2 Hardware Interfaces

Since the application must run over the Android OS and over the internet, all the hardware shall require to connect internet will be hardware interface for the system.

3.1.3 Software Interfaces

1. The system shall communicate with the Configurator to identify all the available components to configure the service.
2. The system shall communicate with the content manager to get the product specifications, offerings and promotions.

3. The system shall communicate with bKash and the DBBL Mobile Banking system to identify available payment methods, validate the payments and process payment.
4. The system shall communicate to credit management system for handling financing options.
5. The system shall communicate with Google Map system for location finding and tracking.

3.1.4 Communications Interfaces

The HandyMan system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

3.2 Licensing Requirements

Not Applicable

3.3 Legal, Copyright, and Other Notices

HandyMan should display the disclaimers and copyright.

3.4 Applicable Standards

It shall be as per the industry standard.

3.5 Supporting Information

Following documents will be referred:

1. Vision document for HandyMan.
2. Use case analysis.
3. Structural models.
4. Behavioral models.
5. Non functional requirements model.
6. Project Plan

3.6 Programming Language

We will use Java as a programming language. Java is a programming language that is used to produce softwares for multiple platforms. When a programmer writes a Java application, the compiled code runs on most operating systems (OS), including Windows, Linux and Mac OS. Java derives much of its syntax from the C and C++ programming languages.

3.7 XML

Extensible Markup Language (XML) is used to describe data. The XML standard is a flexible way to create information formats and electronically share structured data via the public Internet, as well as via corporate networks. XML data is known as self-describing or self-defining, meaning that the structure of the data is embedded with the data, thus when the data arrives there is no need to pre-build the structure to store the data; it is dynamically understood within the XML.

3.8 Implemented Tools and Platform

Every business plan or project comes down to tactics, tools and strategies. If we want to develop and implement a social media marketing strategic plan then we must need those three critical components.

3.9 Web Server

A web server is a system that delivers content or services to end users over the internet. A web server consists of a physical server, server operating system (OS) and software used to facilitate HTTP communication. A web server is also known as an internet server. The most simple definition is that a web server runs a website by returning HTML files over an HTTP connection. This definition may have been true in the early days of the internet, but the line has blurred between websites, web applications and web services, etc. For example, a server that delivers an XML document to another device can be a web server. A better definition might be that a web server is any internet server that responds to HTTP requests to deliver content and services.

3.10 Database Server

The term database server may refer to both hardware and software used to run a database, according to the context. As software, a database server is the back-end portion of a database application, following the traditional client-server model. This back-end portion is sometimes called the instance. It may also refer to the physical computer used to host the database. We will use MYSQL database server to store all the information of this system.

4. Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well
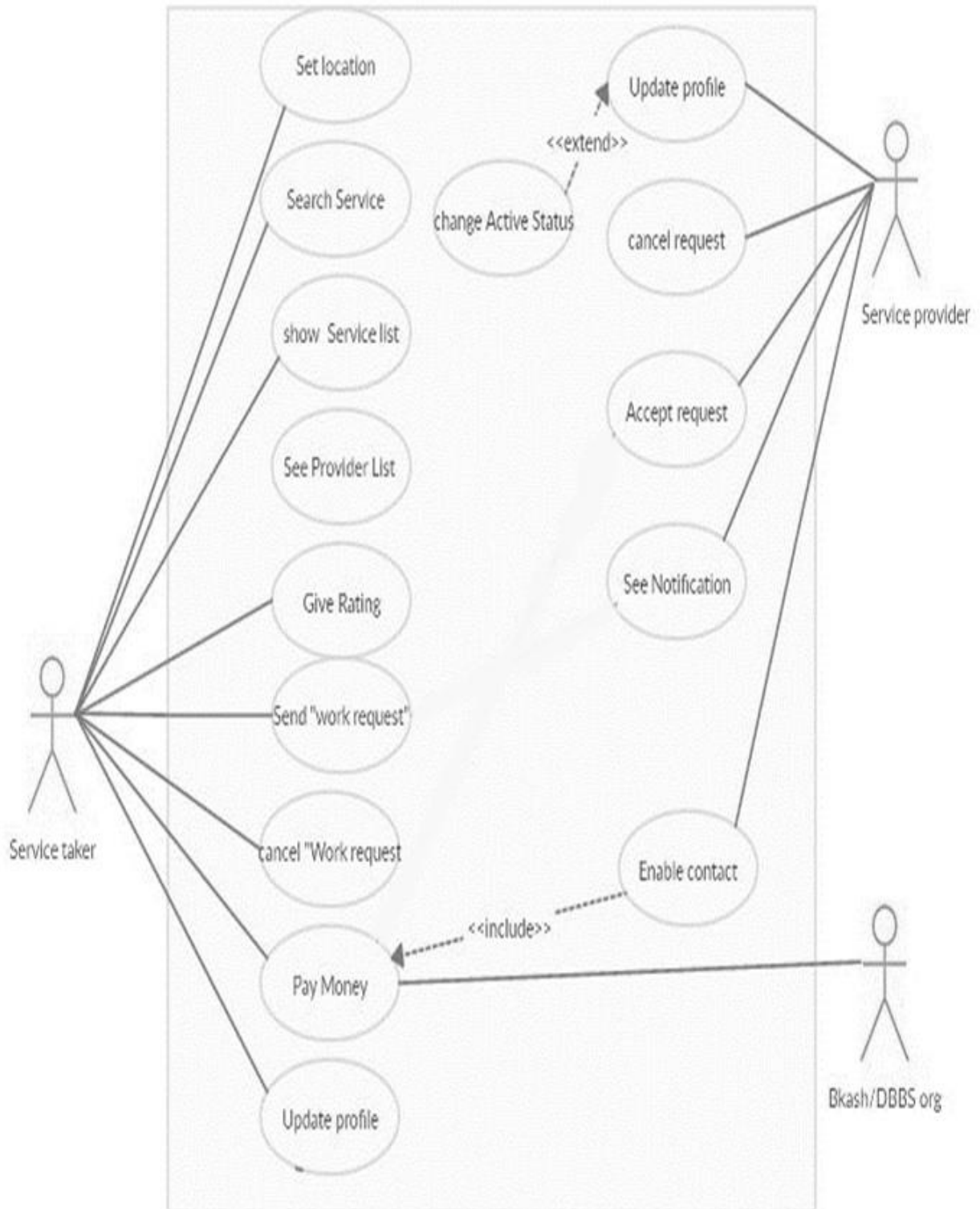
Figure 1: Use Case Diagram of HandyMan

5. Requirement Specification

The complete requirement specification based on the elicitation process is described in this section

5.1 Functional Requirements

Functional requirements are specified something the system should do. It specifies a behavior or function. Here are our Functional requirements.

5.1.1 Service taker sets the Location

R1	Service taker sets Location
Description	In the opening of HandyMan, service taker may need to set location. It would help to find near location of service provider, besides service taker can find his location. Our system will provide him/her a near location available service provider
Stakeholders	Service taker
Priority	High Priority

Table 1: Service taker sets the Location

5.1.2 Service taker Search a Service

R2	Service taker search for service
Description	Service taker can search any type of service what he/she looking for
Stakeholders	Service taker
Priority	High Priority

Table 2: Service taker Search a Service

5.1.3 Service taker choose his/her services

R3	Service taker Choose his/her services
Description	After searching, Service taker sees a multiple Service list. He or she can choose a service or multiple service
Stakeholders	Service taker
Priority	High Priority

Table 3: Service taker choose his/her services

5.1.4 Service taker sees service provider lists & profile

R4	Service taker Sees service provider Lists & profile
Description	Service taker choose his service. then system show Some of his service related service provider name. Service taker can see any of them profile
Stakeholders	Service taker
Priority	High Priority

Table 4: Service taker Sees service provider Lists & profile

5.1.5 Service taker sends "Work Request" to service provider

R5	Service taker sends "work Request" to service taker
Description	Service taker can see any service provider profile so he /she choose service provider for his/her service and sends "work request"
Stakeholders	Service taker
Priority	High Priority

Table 5: Service taker sends "Work Request" to service provider

5.1.6 Service taker cancels "Work Request"

R6	Service taker cancels "work request"
Description	After sending "work request", the service provider does not response or service taker does not want the service at the moment. So, he/she can cancel the "work request"
Stakeholders	Service taker
Priority	High Priority

Table 6: Service taker cancels "Work Request"

5.1.7 Service taker gives rating to service provider

R7	Service taker gives rating to service provider
Description	Service taker can give rating to service provider who is done his/her service. He /She can give ratings 1 to 5 star
Stakeholders	Service taker
Priority	High Priority

Table 7: Service taker gives rating to service provider

5.1.8 Service taker update own profile

R8	Service taker update own profile
Description	Service taker has own profile. He/she can change his/her information like name, email, Mobile no etc.
Stakeholders	Service taker
Priority	Medium Priority

Table 8: Service taker update own profile

5.1.9 Service provider Accept "Work Request"

R9	Service provider Accept "Work request"
Description	Service taker sends work request to Service provider. Service Provider can accept the request if she/he wants.
Stakeholders	Service provider
Priority	High Priority

Table 9: Service provider Accept "Work Request"

5.1.10 Service provider cancels "Work Request"

R10	Service provider cancels "work request"
Description	Service provider can cancel the service taker's "work request if he/she doesn't like
Stakeholders	Service provider
Priority	High Priority

Table 10: Service provider cancels "Work Request"

5.1.11 Service provider updates own profile

R11	Service provider updates own profile
Description	Service provider has own profile. He/she can change his/her information like name, email, Mobile no etc.
Stakeholders	Service provider
Priority	Medium Priority

Table 11: Service provider updates own profile

5.1.12 Service provider changes Active Status

R12	**Service provider changes Active Status**
Description	Service provider can his/her Active status using turn on or off
Stakeholders	Service provider
Priority	Low priority

Table 12: Service provider changes Active Status

5.1.13 Data retrieve from cloud server

R13	**Data retrieve from cloud server**
Description	Data must be retrieved from server as the whole system will be dynamic. It is also to be said that, all operational functionality will be occurred on server also.
Stakeholders	Service provider, service taker
Priority	High priority

Table 13: Data retrieve from cloud server

5.1.14 Service taker pays money

R14	**Service taker pays money**
Description	After service provider accepts the request, the service taker pays money for his/her service using bkash/DBBLs.
Stakeholders	Service provider, service taker
Priority	High priority

Table 14: Service taker pays money

5.2 Data Requirements:

Data requirements are prescribed directives or consensual agreements that define the content and/or structure that constitute high quality data instances and values. Here are requirements for Data:

- The system shall input, process and output data types of integers, strings, characters and real due to the variation of information that will be stored

- The system shall have accurate and up-to date information

- The system shall handle information from various sources

Firstly, we need build a data model. We need to focus on entity of data, quantity of data, capacity of data resource, data availability etc. For collecting data, we are following these steps:

- **Data Object Selection:** A data object is a representation of information which has different properties or attributes that must be understood by software. Firstly we find all Attributes of our System.
- **Data Objects and Attributes:** we find necessary attribute for each Entity
- **Relation between Data object:** Now we define all relation between two entity and it's attributes.
- **E-R Diagram:** Entity relationship diagram displays the relationships of entity set stored in a database.
- **Schema Diagram:** Using E-R diagram, we draw Schema of our database
- **Create table:** Now create table and set data types it's attribute.
- **Normalization :** Normalization is the process of reorganizing data in a database so that it meets two basic requirements:
    1. There is no redundancy of data (all data is stored in only one place), and
    2. Data dependencies are logical

  At last, we do Normalization.

5.3 Performances Requirements:

Performances are very Important issue of our application. Performance requirements define how well the system performs certain functions under specific conditions. The Performance requirements are often based on Supporting our System end User Tasks. To ensure our System Performance, we need to maintain some issue. Here we describe the issues by which we are going to enhance the performance of our Application.

5.3.1 Speed & Latency Requirements:

Speed & Latency requirements must be ensured while retrieving data from the cloud Server

SLR-1	Search must be faster
Description	When service taker search for service, the search result must show within one second
Stakeholder	Service Taker
Priority	High Priority

Table 15: Search must be faster

5.3.2 Precision & Accuracy Requirements:

Search result that is shown to end user Must be accurate. Wrong Information might be Ruined the whole business Process.

PAR-1	Search Result must be Accurate
Description	When Service taker search for service, the result must be According to input value given by service taker
Stakeholder	Service Taker
Priority	High Priority

Table 16: Search Result must be Accurate

5.3.3 Capacity Requirements:

The system must be capable to handle user data, provide accurate data, handling database, manage http etc.

CR-1	The system Must be handled 100 thousands of data
Description	The system needs to handle 100 thousands of every moment
Stakeholder	Service taker, service provider
Priority	High Priority

Table 17: The system must be handled 100 thousands of data

5.4 Dependability Requirements:

The systems have the ability to deliver the service when it's requested, the ability to deliver the service as specified, the ability to operate without catastrophic failure, the ability to protect itself against accidental intrusion, the ability of to resist or recover from damaging event. So the term Dependability measures as five dimensions such as

- Availability

- Reliability

- Safety

- Security

- Resilience

5.4.1 Reliability & Availability Requirements:

RAR-1	The system must be Available on 24X7
Description	The system must be available 24 hours, every day in a week
Stakeholder	Service taker, service provider
Priority	High Priority

Table 18: The system must be Available on 24X7

5.4.2 Robustness or Fault-Tolerance Requirements:

RFR-1	The system handles all user access without system errors
Description	Thousands of users might hit our application system at a time. All their requests must be handled without any fault.
Stakeholder	Service taker, service provider
Priority	High Priority

Table 19: The system handles all user access without system errors

5.4.3 Safety-Critical Requirements:

There are no safety-critical requirements in our project.

5.5 Maintainability & Supportability Requirements:

It is very important to provide after service or support to the end users.

5.5.1 Maintainability Requirements:

MR-1	System helps to update user profile
Description	In our system, Service taker & service provider can update their own profile
Stakeholder	Service taker, service provider
Priority	Medium Priority

Table 20 : System helps to update user profile

5.5.2 Supportability Requirements:

Supportability requirements may have related to some extends. Like:

- Testability

- Extensibility

- Adaptability

- Maintainability

- Compatibility

- Configurability

- Serviceability

- Install ability

Our application meets all of the above requirements related to supportability

5.5.3 Adaptability Requirements:

There are no adaptability requirements in our system software.

5.6 Security Requirements:

Security Requirements is very important issue for making software. We need to ensure user security and data. Here are our Security requirements:

- Service provider & service taker must sign up

- Email Verification for service provider and service taker

- NID card number Verification for service provider

- Get access only verified and logged user(service taker & service provider)

- Get & change information only logged user(service taker & service provider)

- Password encrypted hashed(Md5) password and store in database and it is one way password

- service taker & service provider can log out to the system

5.6.1 Access Requirements:

For accessing our system, our end user must be verified and authentic. In every module in my Application can Access only Authentic user.

AR-1	Application provides Security Mechanism
Description	In our system, Every Module can access only Authorized and verified User
Stakeholder	Service taker, service provider
Priority	High Priority

Table 21: Application provides Security Mechanism

5.6.2 Integrity Requirements:

Integrity Requirements refers to a security system which ensures an expectation of data quality. In our system, we keep our end user data safely & and never be accidental damaged & exposed data. We keep our end user password and store in database which is encrypted to Hashed password. It is impossible to decrypt to the Hashed Password.

5.6.3 Privacy Requirements:

In our system, we need to ensure our user privacy. There are no third party can access the system without knowing password. User are permitted to access the system to logged user

5.7 Usability and Human-Interaction Requirements:

Our system must be user friendly and anyone can use easily.

5.7.1 Ease of Use Requirements:

Our application is easy to use and also easily understandable.

UHIR-1	Application must be useable for our user
Description	Our App's every modules are understandable and useable anyone can operate our system.
Stakeholder	Service taker, service provider
Priority	High Priority

Table 22: Application must be useable for our user

5.7.2 Personalization and Internationalization Requirements:

There are not any personalization and internationalization requirements to our system. This maiden version of our application is only be operated by Noakhali, Bangladesh.

5.7.3 Understand ability and Politeness Requirements:

Our system is useable enough and Any educated and literate person can understand our every module of our system. We make our system are user friendly. If error occurs, then our system will provide hints or how to solve it, then any one can easily operate our system.

5.7.4 Accessibility Requirements:

There are no specific accessibility requirements associated to our system yet.

5.7.5 User Documentation Requirements:

There are no specific User Documentation requirements for our system

5.7.6 Training Requirements:

Our application is open business Application. Anyone can use our system. Our main stakeholders are service taker, & service provider, so it is impossible to training every of our Stakeholder

5.8 Look and Feel Requirements:

Look & Feel requirements refers how look our Apps user interface and how user expect the user graphic interface.

5.8.1 Appearance Requirements:

Service taker and service provider must know which input fields are required and which are not. For that reason, we will use labels for all input fields. Input fields might be text type, radio, checkbox etc.

AR-1	Labels of mandatory fields must be bold
Description	The mandatory field's label must be bold and all input fields must have placeholder to make it easier for the users
Stakeholder	Service taker, service provider
Priority	Low priority

Table 23: Labels of mandatory fields must be bold

5.8.2 Screen Requirements

Our application can use different type of mobile models. Three are different types models are different size. Our user interface must be fill full of mobile screen

SR-1	Application must fill the full mobile Screen
Description	For Different size of mobile, the user graphical interface automatically changes and fill the full screen
Stakeholder	N/A
Priority	High Priority

Table 24: Application must fill the full mobile Screen

5.8.3 Style Requirements:

After keeping all contents, it is very essential to load stylesheet to the application. For mobile application like android system, extensive mark-up language or xml is used. It is to be said that we are going to develop our system at android platform. Style makes the system lucrative.

STY-1	The appearance must be controllable using stylesheet file
Description	For android application style-sheet files are xml. So, all stylesheet must be controllable by the xml file
Stakeholder	N/A
Priority	Medium Priority

Table 25: The appearance must be controllable using stylesheet file

5.9 Operational and Environmental Requirements:

Operational and environmental requirement refers to the capabilities, performance measurements, process, measurements of effectiveness, measurements of performance, measures of sustainability, measurements of technical performances etc.

5.9.1 Expected Physical Requirements:

There are no specific Expected Physical Requirement for our system.

5.9.2 Requirements for Interfacing with Adjacent Systems:

There are no requirements for Interfacing with Adjacent System for our system.

5.9.3 Release Requirements:

There are no specific Release requirements for our system.

5.10 Legal Requirements:

Legal requirements refer terms of condition, privacy policy etc.

5.10.1 Compliance Requirements:

There are no specific standards requirements for our system.

5.10.2 Standards Requirements:

There are no specific standards requirements for our system.

6. Requirement Engineering Process

Requirements engineering refers to the process of defining, documenting and maintaining requirements in the engineering design process. It is a common role in systems engineering and software engineering.

6.1 Requirement Elicitation Techniques

Requirement elicitation is the process of collecting and refining stakeholder's requirements. Projects are garbage-in-garbage-out meaning that poor quality requirements typically lead to project issues and failure.

6.1.1 Hold Elicitation Interviews

We hold interviews that can be performed one-on-one of stakeholders. They are an effective way to elicit requirements without taking too much stakeholder time because we meet with people to discuss only the specific requirements that are important to this system. Interviews are helpful to separately elicit requirements from members in preparation for workshops where those members of this system come together to resolve any conflicts.

6.1.2 Current System Analysis

Analyzing the current system can help reveal how systems currently work or what they are supposed to do. Documentation includes any written information about current systems, business processes, requirements specifications, competitor research. Reviewing and analyzing the documents can help identify functionality that needs to remain, functionality that isn't used.

6.1.3 Distribute Questionnaires

We arranged an online survey to collect requirements for this project. Questionnaires are a way to survey large groups of users to determine what they need. Questionnaires are useful with any large user population but are particularly helpful with distributed groups.

6.2 Requirement Validation

Validation ensures that the requirements are correct and demonstrate the desired quality that you want from this system. Requirements that seem fine when you read them might turn out to have ambiguities and gaps when to try to work with them.

6.2.1 Review the Requirements

Peer review of requirements, particularly the type of rigorous review called inspection, is one of the highest-value software quality practices available. Assemble a small team of reviewers who represent different perspectives and carefully examine the written requirements, analysis models, and related information for defects.

6.2.2 Test the Requirements

We test constitute an alternative view of the requirements. We also conduct writing tests about how to tell if the expected functionality was correctly implemented. Derive tests from the user requirements to document the expected behaviour of the product under specified conditions.

6.2.3 Simulate the requirements

To simulate the requirements commercial tools are available that we have used to simulate a proposed system either in place of or to augment written requirements specifications. Simulation takes prototyping to the next level.

6.3 Change Management

We used an online survey for handling change requests and tracking open issues is essential. Change always has a price, so using change management practices to control scope creep is vital in a contract-development situation. We will provide these following issues in change management.

- Evaluate and prioritize defect corrections and enhancement requests
- Dynamically adjust the scope of future releases or iterations
- Evaluate the impact of proposed changes on users and business processes participation in making change decisions

Appendix

We've prioritized the requirements by following **Three-level Scale** technique.

Three-level Scale: When a BA categorizes the requirements in any of the ordering or ranking scale, it is subject to the analyst's understanding of the business. Many analysts suggest that this method has some drawbacks and advocate methods that have more than one scale.

Covey, Rebecca and Merrill would have never in their wildest dreams have thought that their "The four-quadrant **'Eisenhower Decision Matrix'** for importance and urgency", from their self-help book First things First, would become one of the most widely used prioritization techniques in the IT space.
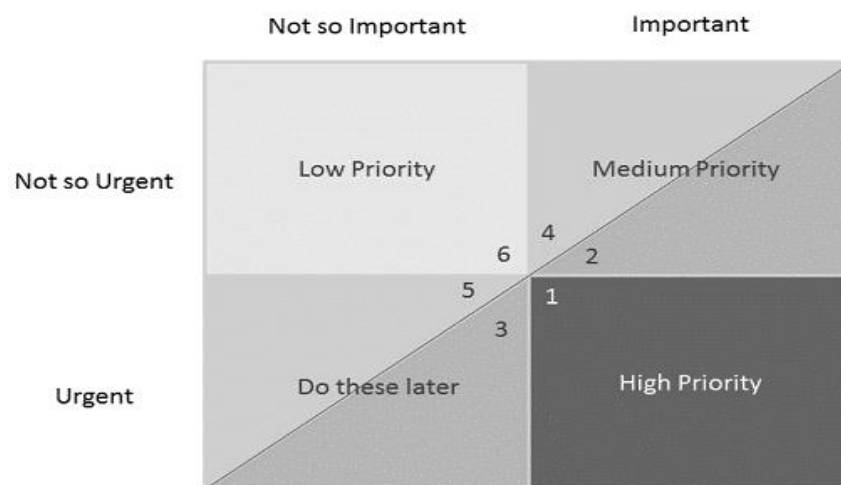

Figure 2: Eisenhower Decision Matrix – Lower the number, higher the priority of the section

.
With the numbering on the different sections of the diagram, the priority of the sections is implicit. Important items have the highest preference, while urgent items have lower preference.

1. High Priority – These requirements are urgent and important. These are requirements that are generally with respect to compliance or contract that cannot be left out. These requirements need to be implemented in the current release and not implementing the same will have some adverse effect on the business.
2. Medium Priority – These requirements are important but not as urgent. Implement these after you implement the high priority items. If you see closely there is a line that splits this quadrant into 2 parts. Implement the items that are on the right side of the line first as they are relatively of higher medium priority.
3. Do these later – These items are urgent but do not have a lot of effect on the business. Hence do it after completing the more important medium priority items. Similar to the

medium priority items, this quadrant has also been split into two; the items on the right side have a higher priority relatively to the items on the left.
4. Low Priority – These items are neither important nor are they urgent. Complete the items at your leisure after completing the items in sections 4 and 5 respectively.

The items on the right-hand side of the diagonal have higher priority. Start with the bottom-right corner of the high-priority quadrant and work your way up and left.

Prioritization of the functional requirements of HandyMan:

R1 – High Priority: Setting Location for the service taker is very essential. Without setting location service taker can't find service provider from near location. So, this requirement is urgent and important.

R2 – High Priority: Searching for service for the service taker is urgent and important. Without searching for service the service taker can't find his desired service.

R3 – High Priority: Without choosing service from the search result service taker can't take any service he needs. So, this requirement is urgent and important.

R4 – High Priority: After choosing any service the service taker needs to see the list of service providers who are working for that service and their work profile so that service taker can choose the best and fit service provider. Thus this requirement is urgent and important.

R5 – High Priority: To hire a service provider sending work request to him is urgent and important. Without sending work request service taker can't notify any service provider to provide service.

R6 – High Priority: Service taker needs to cancel work request if he finds any service provider better than the previous chosen one or he is not willing to take service on that moment. Thus this requirement is urgent and important.

R7 – High Priority: To find best and fit service provider for a service and to justify a service provider, rating from the service taker is very much important and urgent.

R8 – Medium Priority: Service taker can update his profile information providing his full address and much other introductory information. Sometimes full address may help the service provider to locate that service taker more easily. But it can be done from the mapping and location feature also. So this requirement is not so important and not so urgent.

R9 – High Priority: Service provider needs to accept a work request first to get contact information, to proceed to the payment section for that service and to do the job. So, this requirement is urgent and important.

R10 – High Priority: Sometimes service provider needs to cancel a work request when he is not in a situation to provide that service or for any other problem of the service provider. Thus, this requirement is urgent and important.

R11 – Medium Priority: Service provider may need to change any information of his profile like phone number, address etc. for various reasons. This requirement is useful but not so much important.

R12 – Low Priority: Active status of the service providers let the service takers know whether the service provider is in online or not. It's not so urgent or not so important for the service takers to know active status of the service providers.

R13 – High Priority: Data retrieve is urgent and important too. For whole system dynamic, we need to data retrieve to cloud server.

R14 – High priority: Service taker pays money, it is urgent and important. Because without paying money, service provider and service taker cannot communicate to each other.

Prioritization of the Performance requirements of HandyMan:

SLR1 – High Priority: Search service must be faster. It is urgent & important. Search result should show within a second.

PAR1 – High Priority: Accurate result are urgent. Because wrong information might be running the whole business system.

CR1 – High Priority: The system must be above thousands of data. Because there are above thousands of service taker and service provider. So it is urgent and important.

Prioritization of the Dependability requirements of HandyMan:

RAR1 – High Priority: any Service taker can need any service at any time. So our system must be available at any time. So it is urgent and important

RFR1-High priority: If System error occurs in the Application, then we should handle the error. It is urgent and important.

Prioritization of the Maintainability requirements of HandyMan:

MR1-Midium Priority: Service taker and service provider may need to change any information of his profile like phone number, address etc. for various reason. This requirement is useful but not so much important.

Prioritization of the Usability and Human-Interaction requirements of HandyMan:

UHIR1– High priority: It is urgent and also important. Because system must be understandable and useable.

Prioritization of the Usability and Human-Interaction requirements of HandyMan:

AR1 – Low priority: It is not important that label fields are bold.

SR1 – High priority: Our system can use different type mobile models. So user interface must be filling full mobile Screen. So it is Urgent and important

STY1 – Medium Priority: It is important that stylesheet files are xml. So all stylesheet must be controllable be the xml file.