# Use Case Diagram, Activity Diagram and RTM
For

# 'HandyMan'

**Prepared By**

Abdullah An-Noor (ASH1825001M)
Fazle Rabbi (ASH1825004M)
Mahfujur Rahman (ASH1825024M)
Foyzul Karim Pathan (ASH1825033M)

**Project Supervisor**

Md Iftekhar Alam Efat
Lecturer,
IIT, NSTU

BSc. In Software Engineering
Institute of Information Technology
Noakhali Science and Technology University

# Table of Contents

# List of Figures

## 1.1 Use Cases

A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

## 1.2 Benefits of Use Cases

Use cases add value because they help explain how the system should behave and in the process, they also help brainstorm what could go wrong. They provide a list of goals and this list can be used to establish the cost and complexity of the system. Project teams can then negotiate which functions become requirements and are built.

| Use Cases Include | Use Cases Don't Include |
|---|---|
| • Who is using the website<br>• What the user want to do<br>• The user's goal<br>• The steps the user takes to accomplish a particular task<br>• How the website should respond to an action | • Implementation-specific language<br>• Details about the user interfaces or screens. |

## 1.3 Elements of a Use Case

Depending on how in depth and complex you want or need to get, use cases describe a combination of the following elements:

- **Actor** – anyone or anything that performs a behavior (who is using the system)
- **Stakeholder** – someone or something with vested interests in the behavior of the system under discussion (SUD)
- **Primary Actor** – stakeholder who initiates an interaction with the system to achieve a goal
- **Preconditions** – what must be true or happen before and after the use case runs.
- **Triggers** – this is the event that causes the use case to be initiated.
- **Main success scenarios** [Basic Flow] – use case in which nothing goes wrong.
- **Alternative paths** [Alternative Flow] – these paths are a variation on the main theme. These exceptions are what happen when things go wrong at the system level.

## 1.4 Use Case Diagram of ‘HandyMan’

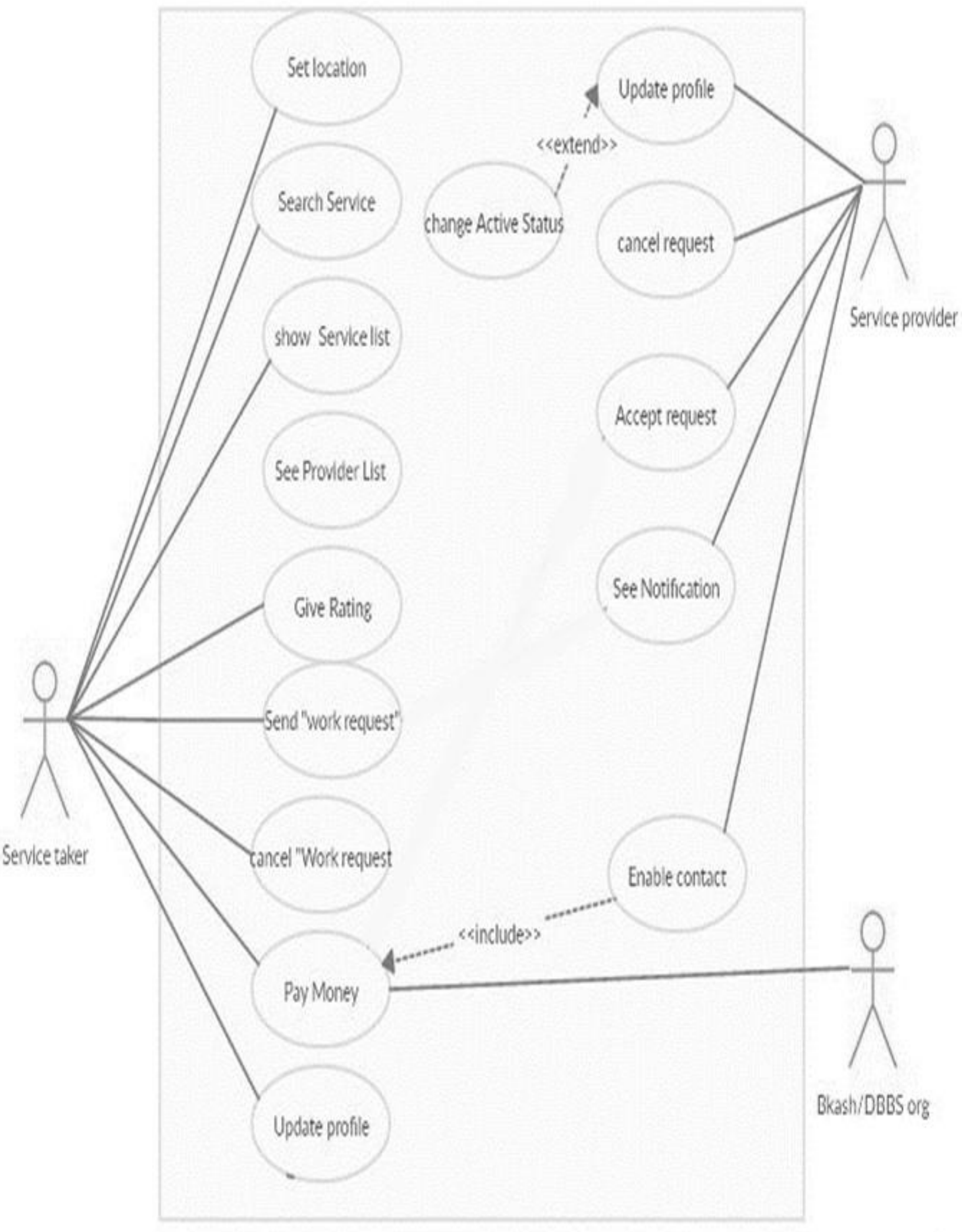


*Figure 1: Use Case Diagram for HandyMan*

# 1.5 Use Case's Description of 'HandyMan'

### Table No. 1.1 | Use Case – 1 : Set Location | Use Case ID : uc_1

<table>
<tr><td><b>Use case</b></td><td><b>Set Location</b></td></tr>
<tr><td><b>Goal</b></td><td>Service taker needs to set location to get service provider on his area and his nearby locations</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Showing searched and nearby locations</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Typing in search location option</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td><b>Step</b></td><td><b>Action</b></td></tr>
<tr><td>1</td><td>Typing in search location option</td></tr>
<tr><td>2.1<br>2.2</td><td>Service taker clicks enter<br>Service taker clicks 'search' button</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>N/A</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.2 | Use Case – 2 : Search Service | Use Case ID : uc_2

<table>
<tr><td><b>Use case</b></td><td><b>Search Service</b></td></tr>
<tr><td><b>Goal</b></td><td>Service taker can search any type service search and requested for service</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Showing searching services</td></tr>
<tr><td><b>Failed End Condition</b></td><td>N/A</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Typing in search option</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td><b>Step</b></td><td><b>Action</b></td></tr>
<tr><td>1</td><td>Typing in search option</td></tr>
<tr><td>2.1<br>2.2</td><td>Service taker clicks enter<br>Service taker clicks 'search' button</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>N/A</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.3 | Use Case – 3 : Select service | Use Case ID : uc_3

<table>
<tr><td><b>Use case</b></td><td><b>Select Service</b></td></tr>
<tr><td><b>Goal</b></td><td>Service taker able to choose service</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service taker able to choose any service</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service taker can't choose any service</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker, Service provider<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select a "available Category" service</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><td><b>Step</b></td><td><b>Action</b></td></tr>
<tr><td>1</td><td>Select a "available Category" service</td></tr>
<tr><td>2</td><td>Choose a service</td></tr>
</table></td></tr>
</table>

<table>
<tr><td><b>Alternative Flows</b></td><td><table><tr><th>Step</th><th>Branching Action</th></tr><tr><td>2a</td><td>Select multiple service</td></tr></table></td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.4 | Use Case – 4 : Select Service Provider | Use Case ID : uc_4

<table>
<tr><td><b>Use case</b></td><td><b>Select Service Provider</b></td></tr>
<tr><td><b>Goal</b></td><td>Service taker sees his/her service category service provider list</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service taker sees service provider list</td></tr>
<tr><td><b>Failed End Condition</b></td><td>There is no service provider for the service</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select a “available Category” service</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Select a “available Category” service</td></tr><tr><td>2</td><td>Choose a service</td></tr><tr><td>3</td><td>Select a “service’</td></tr><tr><td>4</td><td>Clicks “next”</td></tr><tr><td>5</td><td>Show the service provider list for the Service Category</td></tr><tr><td>6</td><td>Visits a service provider’s profile</td></tr></table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table><tr><th>Step</th><th>Branching Action</th></tr><tr><td>3a</td><td>Select multiple service</td></tr></table></td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.5 | Use Case – 5 : Send ‘Work Request’ | Use Case ID : uc_5

<table>
<tr><td><b>Use case</b></td><td><b>Send ‘work request’</b></td></tr>
<tr><td><b>Goal</b></td><td>Service taker requested for work to service provider</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service provider receives a ‘work request’</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service provider don’t receive a ‘work request’</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker, Service provider<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select a “available Category” service</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Select a “available Category” service</td></tr><tr><td>2</td><td>Choose a service</td></tr><tr><td>3</td><td>Select a “service’</td></tr><tr><td>4</td><td>Clicks ‘next’</td></tr><tr><td>5</td><td>Choose a ‘service” provider</td></tr><tr><td>6</td><td>Sends a “work request” to service provider</td></tr><tr><td>7</td><td>Service provider receive a “work request</td></tr></table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table><tr><th>Step</th><th>Branching Action</th></tr><tr><td>2a</td><td>Choose multiple service</td></tr></table></td></tr>
<tr><td><b>Quality Requirements</b></td><td><table><tr><th>Step</th><th>Requirements</th></tr><tr><td>6</td><td>The “work request” pop up message will stay no longer than 90 seconds after it is sent by the Service taker</td></tr></table></td></tr>
</table>

### Table No. 1.6 | Use Case – 6 : Cancel ‘Work Request’ | Use Case ID : uc_6

<table>
<tr><td>Use case</td><td colspan="2">Cancel “work request”</td></tr>
<tr><td>Goal</td><td colspan="2">Service provider denied the sender’s request which sends service taker</td></tr>
<tr><td>Preconditions</td><td colspan="2">Service taker sends “work request”</td></tr>
<tr><td>Success End Condition</td><td colspan="2">Service provider confirm that he is not able to work it</td></tr>
<tr><td>Failed End Condition</td><td colspan="2">Service provider accept the work request that he is not able to work it</td></tr>
<tr><td>Primary Actors:<br>Secondary Actors:</td><td colspan="2">Service taker, Service provider<br>N/A</td></tr>
<tr><td>Trigger</td><td colspan="2">Select a “available Category” service</td></tr>
<tr><td rowspan="9">Main Success Flows</td><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Select a “available Category” service</td></tr>
<tr><td>2</td><td>Choose a service</td></tr>
<tr><td>3</td><td>Select a “service’</td></tr>
<tr><td>4</td><td>Clicks ‘next’</td></tr>
<tr><td>5</td><td>Choose a ‘service” provider</td></tr>
<tr><td>6</td><td>Sends a “work request” to service provider</td></tr>
<tr><td>7</td><td>Service provider receive a “work request</td></tr>
<tr><td>8</td><td>Service provider “cancel” the request</td></tr>
<tr><td rowspan="3">Alternative Flows</td><td>Step</td><td>Branching Action</td></tr>
<tr><td>2a</td><td>Choose multiple service</td></tr>
<tr><td>8a</td><td>Service provider Accept the request</td></tr>
<tr><td rowspan="2">Quality Requirements</td><td>Step</td><td>Requirements</td></tr>
<tr><td>6</td><td>The “work request” pop up message will stay no longer than 90 seconds after it is sent by the Service taker</td></tr>
</table>

### Table No. 1.7 | Use Case – 7 : Pay Money | Use Case ID : uc_7

<table>
<tr><td>Use case</td><td colspan="2">Pay Money</td></tr>
<tr><td>Goal</td><td colspan="2">For service, service taker billed for his order work</td></tr>
<tr><td>Preconditions</td><td colspan="2">Service provider Accept “work request”</td></tr>
<tr><td>Success End Condition</td><td colspan="2">Service taker paying the money</td></tr>
<tr><td>Failed End Condition</td><td colspan="2">Service taker not paying the money</td></tr>
<tr><td>Primary Actors:<br>Secondary Actors:</td><td colspan="2">Service taker, Service provider<br>Bkash or DBBL org.</td></tr>
<tr><td>Trigger</td><td colspan="2">Select a “available Category” service</td></tr>
<tr><td rowspan="10">Main Success Flows</td><td>Step</td><td>Action</td></tr>
<tr><td>1</td><td>Select a “available Category” service</td></tr>
<tr><td>2</td><td>Choose a service</td></tr>
<tr><td>3</td><td>Select a “service’</td></tr>
<tr><td>4</td><td>Clicks ‘next’</td></tr>
<tr><td>5</td><td>Choose a ‘service” provider</td></tr>
<tr><td>6</td><td>Sends a “work request” to service provider</td></tr>
<tr><td>7</td><td>Service provider receive a “work request</td></tr>
<tr><td>8</td><td>Service provider accept the request</td></tr>
<tr><td>9<br>9.1<br>9.2</td><td>Service taker paying bill<br>Using bkash<br>Using DBBL</td></tr>
<tr><td rowspan="3">Alternative Flows</td><td>Step</td><td>Branching Action</td></tr>
<tr><td>2a</td><td>Choose multiple service</td></tr>
<tr><td>8a</td><td>Service provider cancel the request</td></tr>
<tr><td rowspan="2">Quality Requirements</td><td>Step</td><td>Requirements</td></tr>
<tr><td>6</td><td>The “work request” pop up message will stay no longer than 90 seconds after it is sent by the Service taker</td></tr>
</table>

### Table No. 1.8 | Use Case – 8 : Give Rating | Use Case ID : uc_8

<table>
<tr><th>Use case</th><th>Give Rating</th></tr>
<tr><td><b>Goal</b></td><td>Service taker can give rating to service Provider</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service taker can give rating to service Provider</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service taker doesn’t give with service provider</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select “My work” in the home page</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select “My work” in the home page</td></tr>
<tr><td>2</td><td>Then click “Completed work”</td></tr>
<tr><td>3</td><td>Click “Service provider’s profile</td></tr>
<tr><td>4<br>4.1<br>4.2<br>4.3<br>4.4</td><td>In the profile, you see profile and give rating click “1Star”<br>Clicking “2Star”<br>Clicking “3Star”<br>Clicking “4Star”<br>Clicking “5star”</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>N/A</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.9 | Use Case – 9 : Update Profile | Use Case ID : uc_9

<table>
<tr><th>Use case</th><th>Update profile</th></tr>
<tr><td><b>Goal</b></td><td>Service taker can update profile like name, birthdate, location etc.</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service taker updates information of his/her own.</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service taker doesn’t update profile successfully.</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select “My Profile” in the home page</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select “My Profile” in the home page</td></tr>
<tr><td>2</td><td>Then “click” edit</td></tr>
<tr><td>3a.<br>3b<br>3c</td><td>Type name<br>Type birthdate<br>Type address</td></tr>
<tr><td>4</td><td>Then click “save”</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>N/A</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.10 | Use Case – 10 : See Notification | Use Case ID : uc_10

| Use case | See notification |
|---|---|
| **Goal** | Service taker can See notification |
| **Preconditions** | N/A |
| **Success End Condition** | Service taker able to see notification |
| **Failed End Condition** | Service taker doesn’t able to see notification |
| **Primary Actors:**<br>**Secondary Actors:** | Service taker<br>N/A |
| **Trigger** | Select “Notification” in the home page |

<table>
<tr><td><b>Main Success Flows</b></td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Select “Notification” in the home page</td></tr></table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>N/A</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.11 | Use Case – 11 : Accept ‘work request’ | Use Case ID : uc_11

<table>
<tr><td><b>Use case</b></td><td><b>Accept “work request”</b></td></tr>
<tr><td><b>Goal</b></td><td>Service provider accepted the sender’s request which sends service taker</td></tr>
<tr><td><b>Preconditions</b></td><td>Service provider sends “work request”</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service provider confirm that he is able to work it</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service provider cancel the work request that he is not able to work it</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker, Service provider<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select a “available Category” service</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table><tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select a “available Category” service</td></tr>
<tr><td>2</td><td>Choose a service</td></tr>
<tr><td>3</td><td>Select a “service’</td></tr>
<tr><td>4</td><td>Clicks ‘next’</td></tr>
<tr><td>5</td><td>Choose a ‘service” provider</td></tr>
<tr><td>6</td><td>Sends a “work request” to service provider</td></tr>
<tr><td>7</td><td>Service provider receive a “work request</td></tr>
<tr><td>8</td><td>Service provider accept the request</td></tr></table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table><tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a</td><td>Choose multiple service</td></tr>
<tr><td>8a</td><td>Service provider cancel the request</td></tr></table></td></tr>
<tr><td><b>Quality Requirements</b></td><td><table><tr><th>Step</th><th>Requirements</th></tr>
<tr><td>6</td><td>The “work request” pop up message will stay no longer than 90 seconds after it is sent by the Service taker</td></tr></table></td></tr>
</table>

### Table No. 1.12 | Use Case – 12 : Accept ‘request’ | Use Case ID : uc_12

<table>
<tr><td><b>Use case</b></td><td><b>Accept “Request”</b></td></tr>
<tr><td><b>Goal</b></td><td>Service provider accepts the “work request”</td></tr>
<tr><td><b>Preconditions</b></td><td>Service taker sends the “work request”</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service provider accept the request</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service provider cancel the work request</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service provider<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select “work Request” in the home page</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table><tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select “work Request” in the home page</td></tr>
<tr><td>2</td><td>Then click ”Accept”</td></tr></table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table><tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a</td><td>Click “cancel</td></tr></table></td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.13 | Use Case – 13 : Cancel “Request” | Use Case ID : uc_13

<table>
<tr><td><b>Use case</b></td><td><b>Cancel “Request”</b></td></tr>
<tr><td><b>Goal</b></td><td>Service provider cancels the “work request”</td></tr>
<tr><td><b>Preconditions</b></td><td>Service taker sends the “work request”</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service provider cancel the request</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service provider accept the work request</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service provider<br>N/A</td></tr>
<tr><td><b>Trigger</b></td><td>Select “work Request” in the home page</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select “work Request” in the home page</td></tr>
<tr><td>2</td><td>Then click ”cancel”</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td><table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a</td><td>Click “Accept”</td></tr>
</table></td></tr>
</table>

### Table No. 1.14 | Use Case – 14 : Show Notification | Use Case ID : uc_14

<table>
<tr><td><b>Use case</b></td><td><b>Show notification</b></td></tr>
<tr><td><b>Goal</b></td><td>Service provider can See notification</td></tr>
<tr><td><b>Preconditions</b></td><td>N/A</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service provider able to see notification</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service provider doesn’t able to see notification</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service provider</td></tr>
<tr><td><b>Trigger</b></td><td>Select “Notification” in the home page</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select “Notification” in the home page</td></tr>
</table></td></tr>
<tr><td><b>Alternative Flows</b></td><td>N/A</td></tr>
<tr><td><b>Quality Requirements</b></td><td>N/A</td></tr>
</table>

### Table No. 1.15 | Use Case – 15 : Enable Contact | Use Case ID : uc_15

<table>
<tr><td><b>Use case</b></td><td><b>Enable Contact</b></td></tr>
<tr><td><b>Goal</b></td><td>Service taker can communicate with service provider</td></tr>
<tr><td><b>Preconditions</b></td><td>Service taker paying the money</td></tr>
<tr><td><b>Success End Condition</b></td><td>Service taker can communicate with service provider</td></tr>
<tr><td><b>Failed End Condition</b></td><td>Service taker can not communicate with service provider</td></tr>
<tr><td><b>Primary Actors:</b><br><b>Secondary Actors:</b></td><td>Service taker, Service provider<br>Credit card Company, bank</td></tr>
<tr><td><b>Trigger</b></td><td>Select a “available Category” service</td></tr>
<tr><td><b>Main Success Flows</b></td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select a “available Category” service</td></tr>
<tr><td>2</td><td>Choose a service</td></tr>
<tr><td>3</td><td>Select a “service’</td></tr>
<tr><td>4</td><td>Clicks ‘next’</td></tr>
<tr><td>5</td><td>Choose a ‘service” provider</td></tr>
<tr><td>6</td><td>Sends a “work request” to service provider</td></tr>
<tr><td>7</td><td>Service provider receive a “work request</td></tr>
<tr><td>8</td><td>Service provider accept the request</td></tr>
</table></td></tr>
</table>

| | | |
|---|---|---|
| | 9<br>9.1<br>9.2 | Service taker paying bill<br>Using bKash<br>Using DBBs |
| | 10.1<br>10.2 | Communicate with Call<br>Communicate with message |
| **Alternative Flows** | **Step** | **Branching Action** |
| | 2a | Choose multiple service |
| | 8a | Service provider cancel the request |
| **Quality Requirements** | **Step** | **Requirements** |
| | 6 | The “work request” pop up message will stay no longer than 90 sec |

### Table No. 1.16 | Use Case – 16 : Update Profile | Use Case ID : uc_16

| **Use case** | **Update profile** | |
|---|---|---|
| **Goal** | Service provider can update profile like name, birthdate, location etc. | |
| **Preconditions** | | |
| **Success End Condition** | Service provider updates information of his/her own. | |
| **Failed End Condition** | Service provider doesn’t update profile successfully. | |
| **Primary Actors:**<br>**Secondary Actors:** | Service provider<br>N/A | |
| **Trigger** | Select “My Profile” in the home page | |
| **Main Success Flows** | **Step** | **Action** |
| | 1 | Select “My Profile” in the home page |
| | 2 | Then “click” edit |
| | 3a.<br>3b<br>3c | Type  name<br>Type birthdate<br>Type address |
| | 4 | Then click “save” |
| **Alternative Flows** | N/A | |

# 2. Activity Diagram

## 2.1 Introduction

**Activity diagrams** are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

## 2. 2 Construction of Activity Diagram

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- Ellipses represent actions;
- Diamonds represent decisions;
- Bars represent the start (split) or end (join) of concurrent activities;
- A black circle represents the start (initial node) of the workflow;
- An encircled black circle represents the end (final node).
- Arrows run from the start towards the end and represent the order in which activities happen.

Activity diagrams can be regarded as a form of a structured flowchart combined with a traditional data flow diagram. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

While in UML 1.x, activity diagrams were a specialized form of state diagrams, in UML 2.x, the activity diagrams were reformalized to be based on Petri net-like semantics, increasing the scope of situations that can be modeled using activity diagrams. These changes cause many UML 1.x activity diagrams to be interpreted differently in UML 2.x.

UML activity diagrams in version 2.x can be used in various domains, e.g. in design of embedded systems. It is possible to verify such a specification using model checking technique.

## 2.3 Activity Diagrams of Use Cases of ‘HandyMan’

### Use Case 1 : Set Location

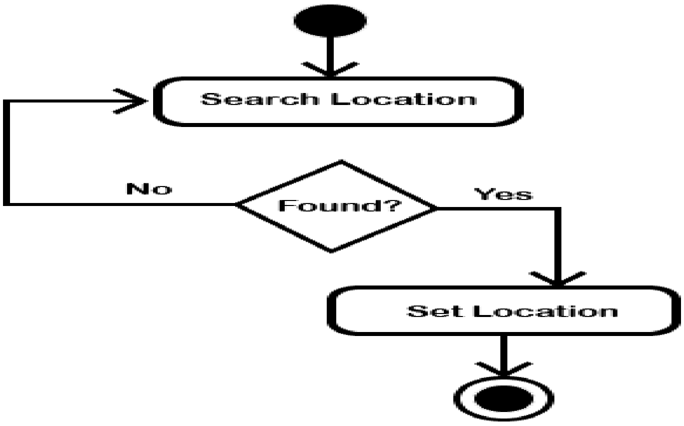


*Figure 2: Activity Diagram for 'Set Location'*

### Use Case 2 : Search Service

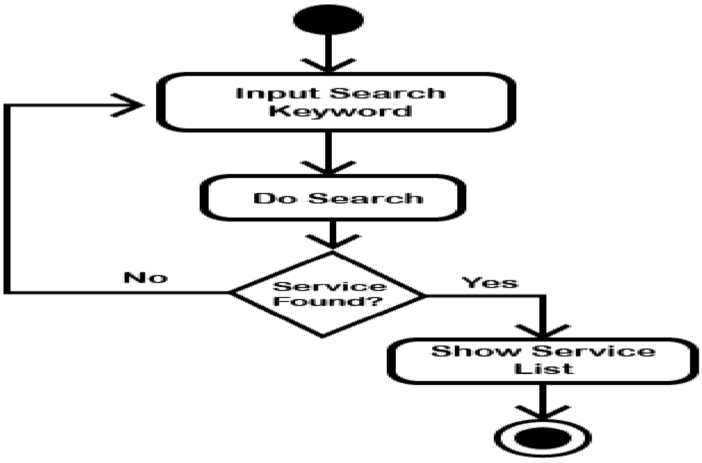


*Figure 3: Activity Diagram for 'Search Service'*

### Use Case 3 : Select Service List

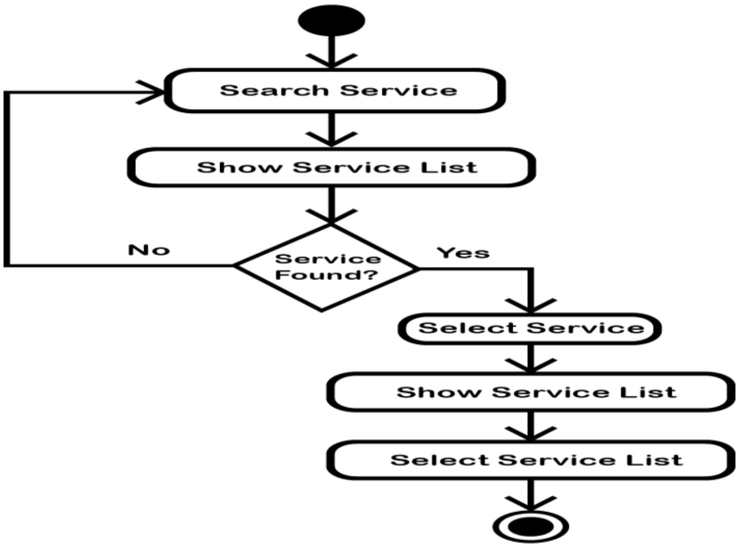


*Figure 4: Activity Diagram for 'Select Service List'*

## Use Case 4 : Select Service Provider

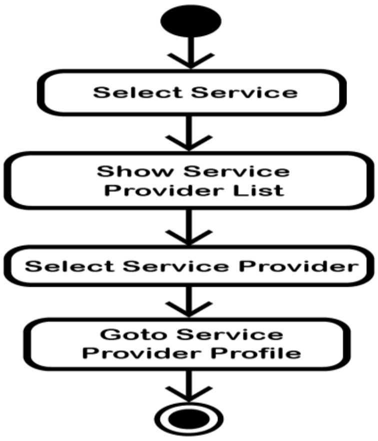


*Figure 5: Activity Diagram for 'Select Service Provider'*

## Use Case 5 : Send Work Request

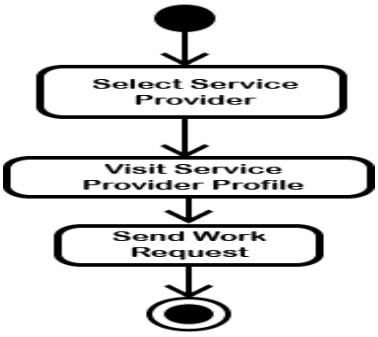


*Figure 6: Activity Diagram for 'Send Work Request'*

## Use Case  6 : Cancel Work Request

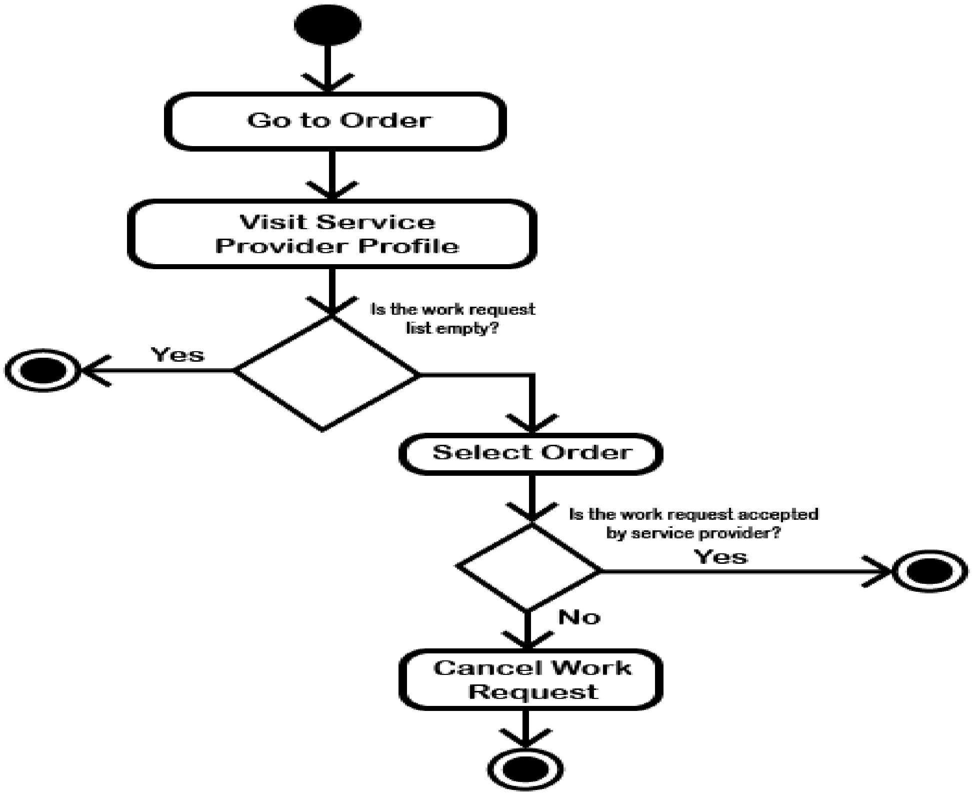

Go to Order
Visit Service Provider Profile
Is the work request list empty?
Yes
Select Order
Is the work request accepted by service provider?
Yes
No
Cancel Work Request

*Figure 7: Activity Diagram for 'Cancel Work Request'*

### Use Case 7 : Pay Money

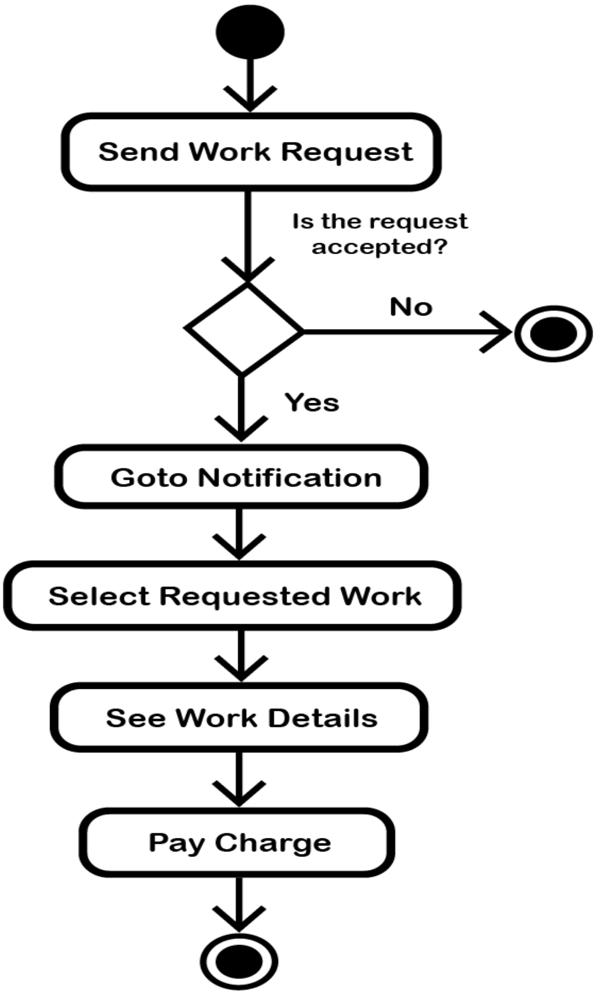


*Figure 8: Activity Diagram for 'Pay Money'*

### Use Case 8 : Give Rating

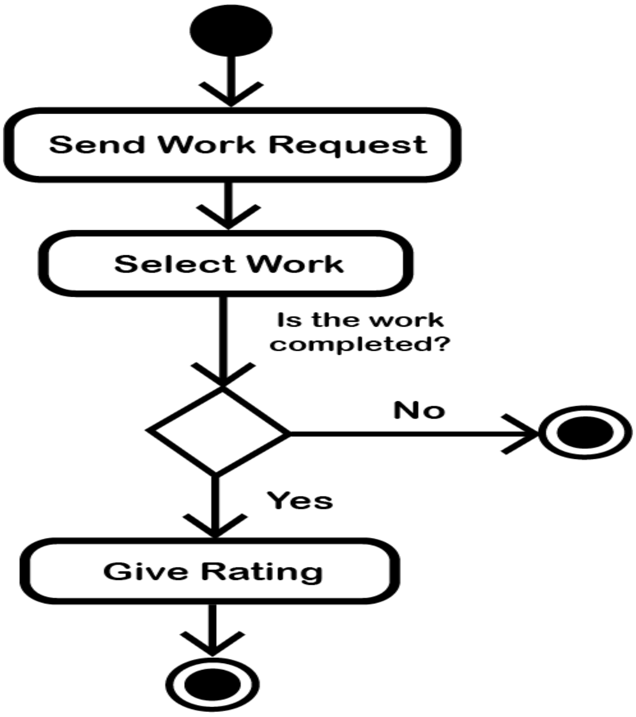


*Figure 9: Activity Diagram for 'Give Rating'*

## Use Case 9 : Update Profile

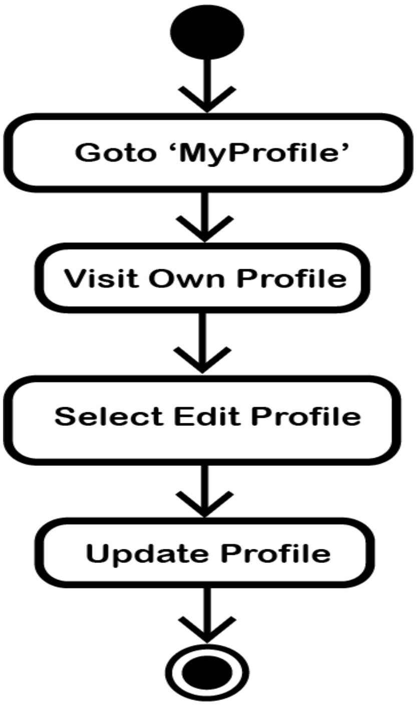


*Figure 10: Activity Diagram for 'Update Profile'*

## Use Case 10 : See Notification

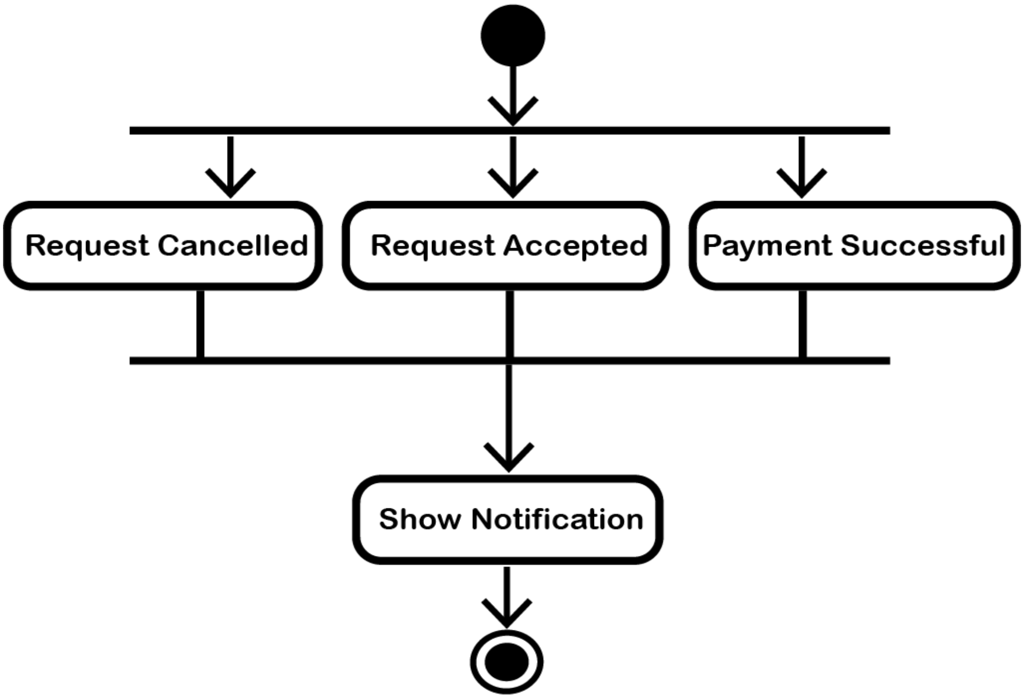


*Figure 11: Activity Diagram for 'See Notification'*

## Use Case 14 : Change Active Status

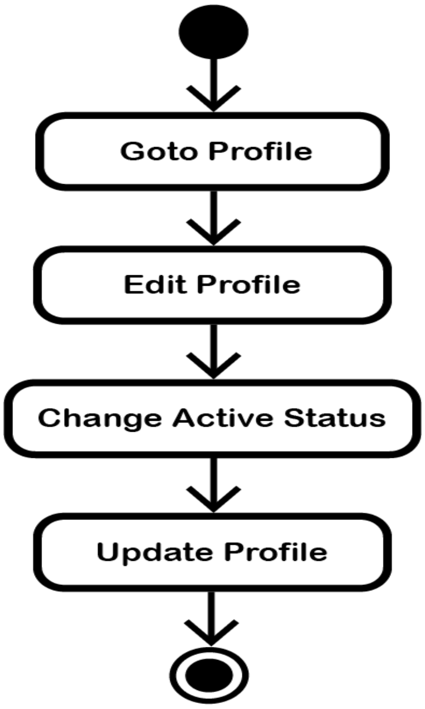


*Figure 12: Activity Diagram for 'Change Active Status'*

## Use Case 15 : Enable Contact

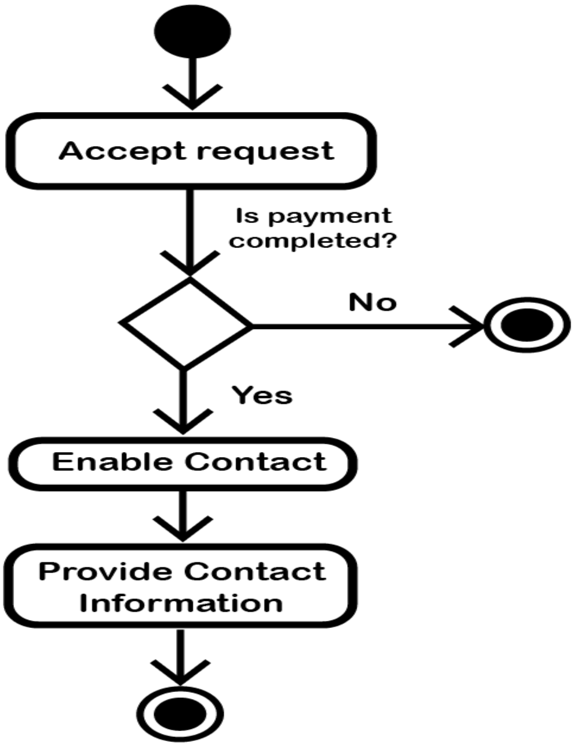


*Figure 13: Activity Diagram for 'Enable Contact'*

# 3. Requirement Traceability Matrix

## 3.1 Introduction

**A Traceability Matrix** is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. It is used to track the requirements and to check the current project requirements are met.

**Requirement Traceability Matrix** (RTM) is a document that maps and traces user requirement with test cases. It captures all requirements proposed by the client and requirement traceability in a single document, delivered at the conclusion of the Software devlopement life cycle. The main purpose of Requirement Traceability Matrix is to validate that all requirements are checked via test cases such that no functionality is unchecked during Software testing.

## 3.2 Why RTM is Important?

The main agenda of every tester should be to understand the client's requirement and make sure that the output product should be defect-free. To achieve this goal, every QA should understand the requirement thoroughly and create positive and negative test cases. This would mean that the software requirements provided by the client have to be further split into different scenarios and further to test cases. Each of this case has to be executed individually. A question arises here on how to make sure that the requirement is tested considering all possible scenarios/cases? How to ensure that any requirement is not left out of the testing cycle? A simple way is to trace the requirement with its corresponding test scenarios and test cases. This merely is termed as 'Requirement Traceability Matrix.' The traceability matrix is typically a worksheet that contains the requirements with its all possible test scenarios and cases and their current state, i.e. if they have been passed or failed. This would help the testing team to understand the level of testing activities done for the specific product.

## 3.3 Which Parameters to include in Requirement Traceability Matrix?

- Requirement ID
- Requirement Type and Description
- Test Cases with Status

**A requirement traceability matrix can:**

- Show the requirement coverage in the number of test cases
- Design status as well as execution status for the specific test case
- If there is any User Acceptance test to be done by the users, then UAT status can also be

  captured in the same matrix.
- The related defects and the current state can also be mentioned in the same matrix.

## Table No. 2 - Business Requirements:

| BR# | Business requirements |
|---|---|
| **BR1** | Allow to service taker to set location |
| **BR2** | Allow to service taker send "work request" to Service provider |
| **BR3** | Allow to service taker to pay money |
| **BR4** | Allow to cancel the work request |
| **BR5** | Allow to service taker to give rating to service provider |
| **BR6** | Allow to update Service taker profile |
| **BR7** | Allow to update service provider profile |

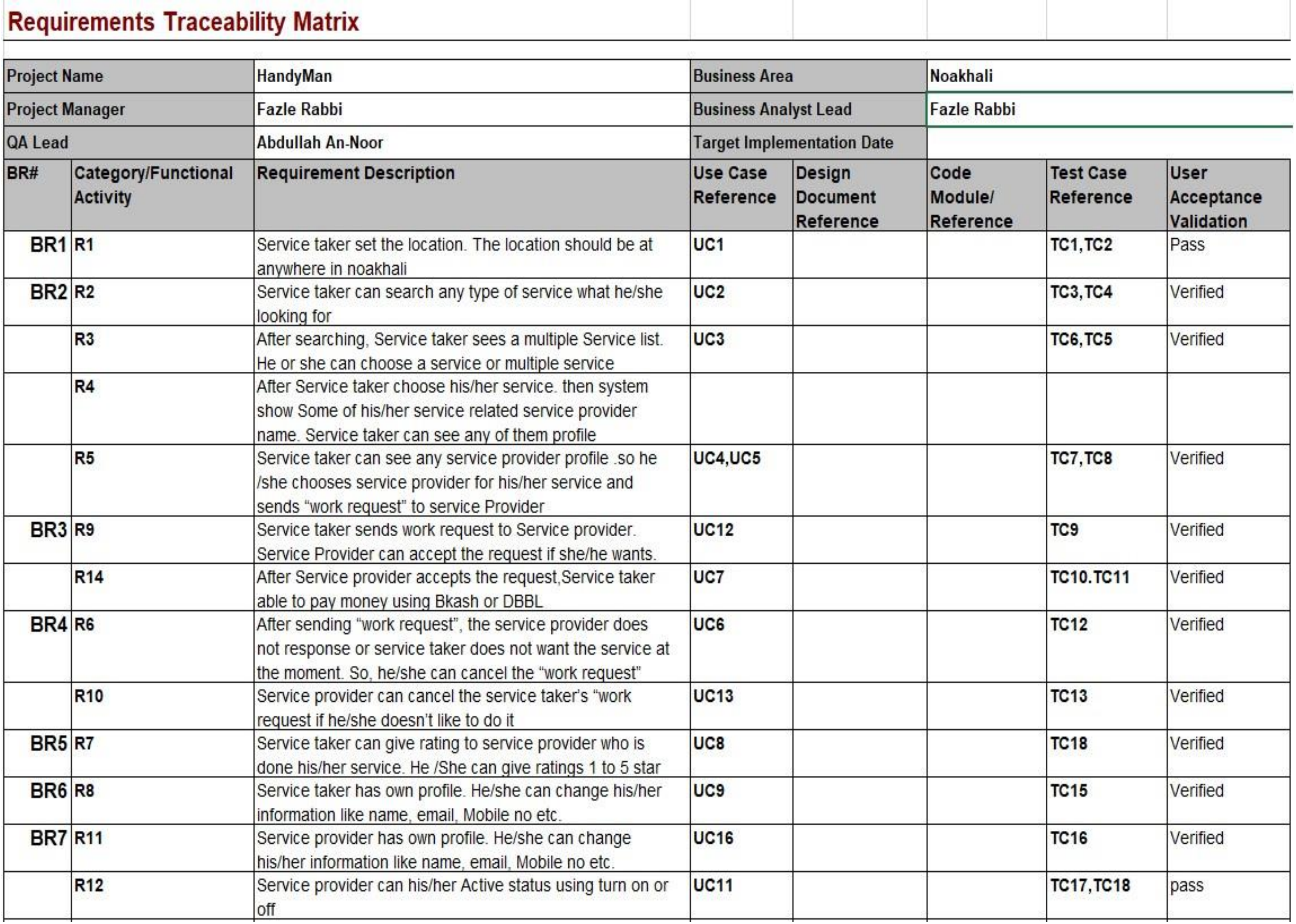
## Table No. 3 - Use Cases:

| **Use Case No** | **Use case name** |
|---|---|
| UC-1 | Set location |
| UC-2 | Search Service |
| UC-3 | Select Service List |
| UC-4 | Select Service Provider |
| UC-5 | Send “Work Request” |
| UC-6 | Cancel “Work request” |
| UC-7 | Pay Money |
| UC-8 | Give Rating |
| UC-9 | Update profile |
| UC-10 | See Notification |
| UC-11 | Change Active Status |
| UC-12 | Accept Request |
| UC-13 | Cancel Request |
| UC-14 | See notification |
| UC-15 | Enable Contact |
| UC-16 | Update Profile |

## Table No. 4 - Test Cases:

| **Test case No** | **Test case** |
|---|---|
| **TC1** | Maijdee Bazar |
| **TC2** | Chowmohoni |
| **TC3** | Ac Servicing |
| **TC4** | Computer Servicing |
| **TC5** | Verify if Service taker able to choose a service |
| **TC6** | Verify if Service taker able to choose multiple Service |
| **TC7** | Verify if service taker is able to choose service provider |
| **TC8** | Verify if service taker is able to send “work request” to service provider |
| **TC9** | Verify if service provider is able to accept the request |
| **TC10** | Verify if service taker is able to pay using bKash |
| **TC11** | Verify if service taker is able to pay using DBBL |
| **TC12** | Verify if service taker is able to cancel the request |
| **TC13** | Verify if service provider is able to cancel the request |
| **TC14** | Verify if service taker is able to give rating to service provider |
| **TC15** | Verify if service taker is able to update information like name, email etc. |
| **TC16** | Verify if service Provider is able to update information like name, email etc. |
| **TC17** | OFF |
| **TC118** | ON |

Table No. 5



## Requirements Traceability Matrix

| | | | |
|---|---|---|---|
| Project Name | HandyMan | Business Area | Noakhali |
| Project Manager | Fazle Rabbi | Business Analyst Lead | Fazle Rabbi |
| QA Lead | Abdullah An-Noor | Target Implementation Date | |

| BR# | Category/Functional Activity | Requirement Description | Use Case Reference | Design Document Reference | Code Module/ Reference | Test Case Reference | User Acceptance Validation |
|---|---|---|---|---|---|---|---|
| BR1 | R1 | Service taker set the location. The location should be at anywhere in noakhali | UC1 | | | TC1,TC2 | Pass |
| BR2 | R2 | Service taker can search any type of service what he/she looking for | UC2 | | | TC3,TC4 | Verified |
| | R3 | After searching, Service taker sees a multiple Service list. He or she can choose a service or multiple service | UC3 | | | TC6,TC5 | Verified |
| | R4 | After Service taker choose his/her service. then system show Some of his/her service related service provider name. Service taker can see any of them profile | | | | | |
| | R5 | Service taker can see any service provider profile .so he /she chooses service provider for his/her service and sends "work request" to service Provider | UC4,UC5 | | | TC7,TC8 | Verified |
| BR3 | R9 | Service taker sends work request to Service provider. Service Provider can accept the request if she/he wants. | UC12 | | | TC9 | Verified |
| | R14 | After Service provider accepts the request,Service taker able to pay money using Bkash or DBBL | UC7 | | | TC10.TC11 | Verified |
| BR4 | R6 | 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" | UC6 | | | TC12 | Verified |
| | R10 | Service provider can cancel the service taker's "work request if he/she doesn't like to do it | UC13 | | | TC13 | Verified |
| BR5 | R7 | Service taker can give rating to service provider who is done his/her service. He /She can give ratings 1 to 5 star | UC8 | | | TC18 | Verified |
| BR6 | R8 | Service taker has own profile. He/she can change his/her information like name, email, Mobile no etc. | UC9 | | | TC15 | Verified |
| BR7 | R11 | Service provider has own profile. He/she can change his/her information like name, email, Mobile no etc. | UC16 | | | TC16 | Verified |
| | R12 | Service provider can his/her Active status using turn on or off | UC11 | | | TC17,TC18 | pass |

## Conclusion

Use Case, Activity Diagram and Requirements Traceability Matrix is very much important for a software requirements specification and analysis. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled. Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Traceability Matrix co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. Without these information, software developers will face promlems and complexity to develop software properly.