

HMI EXPERIMENT 4

Roll No. A-52	Name: JANMEJAY PATIL
Class: BE-A	Batch: A3
Date of Experiment: 09-03-2022	Date of Submission
Grade :	

AIM: Design interface for automated ticket vending machine (ATVM) for any system

B.1 Tools Used to Design the interface:

Sublime Text Editor:

Sublime Text is a commercial source code editor. It natively supports many programming languages and markup languages. Users can expand its functionality with plugins, typically community-built and maintained under free-software licenses. To facilitate plugins, Sublime Text features a Python API.

HTML:

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser.

CSS:

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

JS:

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client-side for web page behavior, often incorporating third-party libraries

B.2 Sample Source code of the application:

INDEX.HTML

```
<!DOCTYPE html>

<html lang="en" ng-app>

  <head>

    <meta charset="utf-8">

    <meta
http-equiv="X-UA-Compatible"
content="IE=edge">

    <meta name="viewport"
content="width=device-width,
initial-scale=1">

    <title>Janmejy Patil</title>

    <!-- Bootstrap -->

    <link
href="bootstrap/css/bootstrap.min.css"
rel="stylesheet">

    <!-- Custom css -->

    <link href="css/atvm.css"
rel="stylesheet">

  </head>

  <body
ng-controller="atvmController">

    <div class="main-container
full-height">

      <div class="container
full-height">

        <div class="panel
panel-default full-height main-panel">

          <div
class="panel-body full-height">
```

```
          <div class="row
set-height">

            <div class="col-sm-3
hidden-xs set-height">

              <h3>All
Stations</h3>

              <hr/>

              <div
class="list-group set-height
scrollable">

                <a href="#"
class="list-group-item"
ng-repeat="station in stations track by
$index"
ng-class="getSelectedStationStyle($index
)"
ng-click="setSelectedStation($index)">{{
station.name}} - {{station.devng}}</a>

              </div>

            </div>

            <div class="col-sm-9
hidden-xs set-height">

              <h3>Choose
Destination</h3>

              <hr/>

              <div
class="btn-group btn-group-justified
btn-group-lg">

                <div
class="btn-group btn-group-lg"
ng-repeat="mainStation in mainStations
track by $index">
```

```

<button
type="button" class="btn btn-default"
ng-class="getSelectedStationStylePrimary
(mainStation)"
ng-click="setSelectedMainStation($index)"
">{{getStation(mainStation)}}

</button>

</div>

</div>

<br/>

<div
class="btn-group btn-group-justified
btn-group-md center-block"
style="width:90%;">

<div
class="btn-group btn-group-md"
ng-repeat="subStation in
getSubStations(selectedMainStation)">

<button
type="button" class="btn btn-default"
ng-class="getSelectedStationStylePrimary
(subStation[1])"
ng-click="setSelectedStation(subStation[
1])">{{subStation[0]}}

</button>

</div>

</div>

<hr/>

<h3>Ticket</h3>

<div
class="ticket center-block">

<div
class="row">

<div
class="col-xs-3">



```

```

</div>

<div
class="col-xs-9">

<h3
class="text-right">Mumbai Metro
Ticket</h3>

</div>

</div>

<hr/>

<div
class="row">

<div
class="col-xs-5">

<h4>Start:

<span
class="text-warning">{{stations[source].
name}} -
{{stations[source].devng}}</span>

</h4>

</div>

<div
class="col-xs-7">

<h3
class="pull-right">Destination:

<span
class="text-success">{{stations[selected
Station].name}} -
{{stations[selectedStation].devng}}</spa
n>

</h3>

</div>

</div>

```

```

<div
class="row">

<div
class="col-xs-4">

<h4>Journey type:</h4>

</div>

<div
class="col-xs-4">

<h3
class="text-primary text-center pointer"
ng-class="getSingleClass()"
ng-click="returnTicket = false">Single
(यात्रा)</h3>

</div>

<div
class="col-xs-4">

<h3
class="text-success text-center pointer"
ng-class="getReturnClass()"
ng-click="returnTicket = true">Return
(वापसी)</h3>

</div>

</div>

<hr/>

<div
class="row">

<div
class="col-xs-6">

<h4
class="text-center">Adults</h4>

</div>

<div
class="col-xs-6">

<h4
class="text-center">Children</h4>

</div>

```

```

</div>

<div
class="row">

<div
class="col-xs-6 text-center">



</div>

<div
class="col-xs-6 text-center">

</div>

<div
class="col-xs-6 text-center">



</div>

<div
class="col-xs-6 text-center">

</div>

<hr/>

<div
class="row">

<div
class="col-xs-5">

```


ATVM.CSS

```
body,
html {
    height: 99%;
    overflow: hidden;

    background-image:
url("../images/background.jpg");

    background-position: center;
    background-size: cover;
}

.main-panel{
    background-color: #ffffff;
    opacity:0.95;
    margin-top: 15px;
}

.grayed {
    background-color: #e3e3e3;
}

.green {
    background-color: #47a447;
    color: #ffffff;
}

.set-height {
    height: 95%;
}
```

```
.full-height {
    height: 99%;
}

.scrollable {
    overflow: auto;
}

.ticket {
    height: 63%;
    width: 95%;
    background-color: #fff0d9;
    border-radius: 20px;
    border-style: dashed;
    border-width: 2px;
    padding: 10px;
}

.pointer {
    cursor: pointer;
}

.opaque {
    opacity: 0.2;
}
```

ATVM-CONTROLLER.JS

```
function num2hindi(num) {

    var hnum = "०१२३४५६७८९";

    var trans = "";

    var copy = num;

    while (copy > 0.5) {

        var temp = copy % 10;

        trans = hnum.charAt(temp) +
trans;

        copy = copy / 10;

    }

    return trans;

}
```

```
function atvmController($scope) {

    $scope.source = 5;

    $scope.noOfAdults = 1;

    $scope.noOfChildren = 0;

    $scope.selectedStation = 0;

    $scope.selectedMainStation = 0;

    $scope.title = "Mumbai Metro";

    $scope.returnTicket = false;

    $scope.range = function(num) {
```

```
        return new Array(num);

    }

    $scope.getSelectedStationStyle =
function(index) {

        if ($scope.source == index) {

            return "grayed";

        } else {

            return "";

        }

    }

    $scope.getSelectedStationStylePrimary =
function(index) {

        if ($scope.source == index) {

            return "grayed";

        } else if
($scope.selectedStation == index) {

            return "btn-success";

        } else {

            return "";

        }

    }

    $scope.getStation = function(index)
{
```

```

        return
$scope.stations[index].name;
    }

    $scope.setSelectedStation =
function(index) {

    for (var i = 0; i <
$scope.mainStations.length; i++) {

        if ($scope.mainStations[i +
1] > index) {

            $scope.selectedMainStation = i;

            break;

        }

    }

    $scope.selectedStation = index;

}

$scope.setSelectedMainStation =
function(index) {

    $scope.selectedMainStation =
index;

    $scope.selectedStation =
$scope.mainStations[index];

}

$scope.getSubstations =
function(index) {

    var startIndex =
$scope.mainStations[index];

    var endIndex = 0;

    if (index + 1 ==
$scope.mainStations.length) {

        endIndex = -1;

```

```

    } else {

        endIndex =
$scope.mainStations[index + 1] - 1;

    }

    //return [startIndex, endIndex];

    if (endIndex == -1) {

        return
[[$scope.stations[startIndex].name,
startIndex]];

    } else {

        var arr = new Array();

        for (var i = startIndex; i
<= endIndex; i++) {

            arr.push([$scope.stations[i].name, i])

        }

        return arr;

    }

}

$scope.getSingleClass = function() {

    if (!$scope.returnTicket)

        return "";

    else

        return "opaque";

}

$scope.getReturnClass = function() {

    if (!$scope.returnTicket)

        return "opaque";

```



```

        else

            return "";

    }

    $scope.getPersonClass =
function(index, number) {

    if (index < number) {

        return "";

    } else {

        return "opaque";

    }

}

$scope.setNoOfAdults = function(num)
{

    $scope.noOfAdults = num;

}

$scope.setNoOfChildren =
function(num) {

    $scope.noOfChildren = num;

}

    $scope.getTotal = function(dest,
ret, ad, ch) {

        var total = 0;

        total =
$scope.stations[$scope.source].costs[des
t];

        if (ret) {

            total = total * 2;

        }

```

```

        total = total * ad + total * ch
* 0.5;

        return total + " (= " +
num2hindi(total) + ")";

    }

    $scope.mainStations = [0, 5, 9, 12,
16, 20];

    $scope.stations = [

        {

            "name": "Borivali",

            "devng": "बोरिवली",

            "costs": [0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

            "time": [0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

        },

        {

            "name": "Kandivali",

            "devng": "कांदिवली",

            "costs": [1, 0, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

            "time": [1, 0, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

        },

        {

            "name": "Malad",

            "devng": "मालाड",

```

```

        "costs": [1, 2, 0, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 0, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

    },

    {

        "name": "Goregaon",

        "devng": "गोरेगाव",

        "costs": [1, 2, 3, 0, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 0, 4, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 2]

    },

    {

        "name": "Jogeshwari",

        "devng": "जोगेश्वरी",

        "costs": [1, 2, 3, 4, 0, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 0, 5,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

    },

    {

        "name": "Andheri",

        "devng": "अंधेरी",

        "costs": [1, 2, 3, 4, 5, 0,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 0,
6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

```

```

    },

    {

        "name": "Vile Parle",

        "devng": "विले पार्ले",

        "costs": [1, 2, 3, 4, 5, 6,
0, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
0, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

    },

    {

        "name": "Santacruz",

        "devng": "सांताक्रुज़",

        "costs": [0, 1, 2, 3, 4, 5,
6, 7, 0, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20],

        "time": [0, 1, 2, 3, 4, 5,
6, 7, 0, 8, 9, 10, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20]

    },

    {

        "name": "Khar Road",

        "devng": "खार रोड",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 0, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 0, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

    },

    {

        "name": "Bandra",

        "devng": "बान्द्रा",

```

```

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 0, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 0, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20]

    },

    {

        "name": "Mahim",

        "devng": "माहिम",

        "costs": [0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 0, 11, 12, 13, 14, 15,
16, 17, 18, 19, 20],

        "time": [0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 0, 11, 12, 13, 14, 15,
16, 17, 18, 19, 2]

    },

    {

        "name": "Matunga",

        "devng": "माहिम",

        "costs": [0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 0, 12, 13, 14, 15,
16, 17, 18, 19, 20],

        "time": [0, 1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 0, 12, 13, 14, 15,
16, 17, 18, 19, 20]

    },

    {

        "name": "Dadar",

        "devng": "दादर",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 0, 13, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 0, 13, 14, 15, 16,
17, 18, 19, 20]

```

```

    },

    {

        "name": "Elphinstone Road",

        "devng": "एल्फिन्स्टन रोड",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 0, 14, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 0, 14, 15, 16,
17, 18, 19, 20]

    },

    {

        "name": "Lower Parel",

        "devng": "लोअर परेल",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 0, 15, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 0, 15, 16,
17, 18, 19, 20]

    },

    {

        "name": "Mahalakshmi",

        "devng": "महालक्ष्मी",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 0, 16,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 0, 16,
17, 18, 19, 2]

    },

    {

        "name": "Mumbai Cent",

        "devng": "मुम्बई सेंट्रल",

```

```

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 0,
17, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 0,
17, 18, 19, 20]

    },

    {

        "name": "Grant Road",

        "devng": "ग्रॅंट रोड",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
0, 18, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
0, 18, 19, 20]

    },

    {

        "name": "Charni Road",

        "devng": "चर्नी रोड",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 0, 19, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 0, 19, 2]

    },

    {

```

```

        "name": "Marine Lines",

        "devng": "मरीन लाइन्स",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 0, 20],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 0, 20]

    },

    {

        "name": "Churchgate",

        "devng": "चर्चगेट",

        "costs": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 0],

        "time": [1, 2, 3, 4, 5, 6,
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 0]

    }

];

$scope.range = function(num) {

    return new Array(num);

}

}

```

B.3 User Interface Designs:

Janmejay Patil

File | E:/academics/Sem%208/HMI/HMI%20Exp%204/index.html

All Stations

Choose Destination

Borivali - बोरिवली

Kandivali - कांदिवली

Malad - मालाड

Goregaon - गोरेगाव

Jogeshwari - जोगेश्वरी

Andheri - अंधेरी

Vile Parle - विले पार्ले

Santacruz - सांताक्रुज

Khar Road - खार रोड

Bandra - बान्द्रा

Mahim - माहिम

Matunga - माहिम

Dadar - दादर

Elphinstone Road - एल्फिन्स्टन रोड

Borivali

Andheri

Bandra

Dadar

Mumbai Cent

Churchgate

Borivali

Kandivali

Malad

Goregaon

Jogeshwari

Ticket

Mumbai Metro Ticket

Start: Andheri - अंधेरी

Destination: Borivali - बोरिवली

Journey type:

Single (यात्रा)

Return (वापसी)

Adults

Children

Total:

Rs. 1 (= ₹)

Print

Janmejay Patil

File | E:/academics/Sem%208/HMI/HMI%20Exp%204/index.html#

All Stations

Choose Destination

Borivali - बोरिवली

Kandivali - कांदिवली

Malad - मालाड

Goregaon - गोरेगाव

Jogeshwari - जोगेश्वरी

Andheri - अंधेरी

Vile Parle - विले पार्ले

Santacruz - सांताक्रुज

Khar Road - खार रोड

Bandra - बान्द्रा

Mahim - माहिम

Matunga - माहिम

Dadar - दादर

Elphinstone Road - एल्फिन्स्टन रोड

Dadar

Elphinstone Road

Lower Parel

Mahalakshmi

Ticket

Mumbai Metro Ticket

Start: Andheri - अंधेरी

Destination: Elphinstone Road - एल्फिन्स्टन रोड

Journey type:

Single (यात्रा)

Return (वापसी)

Adults

Children

Total:

Rs. 65 (= ₹65)

Print

B.4 HMI principles used to design interfaces.

The general principles of the user interface can be given as follows:

i. Aesthetically Pleasing:

- A design is aesthetically pleasing if it is attractive to the eye. It draws attention subliminally, conveying a message clearly & quickly.
- Visual appeal is provided by following the presentation & graphic design principles which include meaningful contrast between screen elements, creating spatial groupings, aligning screen elements, providing three-dimensional representation, & using color & graphics effectively.

ii. Clarity:

- User interface must be clear in visual appearance, concept & wording.
- Visual elements should be understandable & related to real world concepts & functions. Analogies should be simple.
- Interface words & text should be simple, unambiguous, & free of computer jargon.

iii. Compatibility:

Compatibility needs to be provided as

- User Compatibility: "Know the user" is the fundamental principle in interface design as no users are alike & they think, feel & behave differently compared to the developer.
- Task & job compatibility: The structure & flow of functions should permit easy transition between tasks. The user must never be forced to navigate between applications or many screens to complete routine daily tasks.
- Product compatibility: compatibility across products must always be considered in relation to improving interfaces, making new systems compatible with existing systems will take advantage of what users already know & reduce the necessity for new learning.

iv. Comprehensibility:

- The steps to complete a task should be obvious. System should be understandable & flowing in meaningful order.
- A user should know what to look at, what to do, when to do it, where to do it, why to do it & how to do it.

v. Configurability:

- A default configuration as well as easy personalization & customization through configuration and reconfiguration should be provided.
- Customization enhances sense of control, encourages an active role in

understanding & allows personal preferences & differences in experience levels leading to high user satisfaction.

vi.Consistency:

- Consistency is important because it can reduce requirements for human learning by allowing skills learned in one situation to be transferred to another like it.
- Any new system must impose some learning requirements on its uses but avoid unnecessary activity.

vii.Control:

- The user must control the interaction & never be interrupted for errors.
- Actions should result from explicit user requests & be performed quickly.

viii.Directness:

- Tasks should be performed directly & alternatives should be visible reducing the user's mental workload.
- Tasks are performed by directly selecting an object then selecting an action performed & then seeing the action being performed.

ix.Efficiency:

- Transition between various systems controls should flow easily & freely.
- Navigation paths should be as short as possible.
- Eye movement through a screen should be obvious & sequential.

x.Familiarity:

- Build into the interface concepts,terminology,workflows & spatial arrangements already familiar to the user.
- Familiar concepts enable people to get started & become productive quickly.

xi.Flexibility:

- Flexibility is the system's ability to respond to individual differences in people.
- permitting system customization.

xii.Forgiveness:

- People will make mistakes;a system should be able to tolerate those that are common & unavoidable.
- A forgiving system keeps people out of trouble.

xiii. Predictability:

- All actions should lead to results the user expects.Current operations should provide clues as to what will come next.
- Design consistency enhances predictability.

xiv.Recovery:

- A person should be able to retract any action by issuing an undo command.
- The goal is stability or returning easily to the right track when a wrong track has been taken.
- Recovery should be obvious,automatic,easy & natural to perform.

xv.Responsiveness:

- A user must be responded quickly.
- Substantial or more informative feedback is most important for the casual or new system user.
- All requests must be acknowledged in some way.

xvi. Simplicity:

- Simplicity can be achieved by progressive disclosure, provide defaults, minimize screen alignment points, make common actions points,make common actions simpler,& provide uniformity & consistency.

xvii.Transparency:

- Permit the user to focus on the task or job without concerning the mechanics of the interface.
- Working & reminders of workings inside the computer should be invisible to the user.

xviii.Trade-Offs:

- Final design will be based on a series of trade-offs balancing often-conflicting design principles.
- People's requirements always take precedence over technical requirements.

B.5 Target audience of this Interface?

The target audience of this interface will usually be the ones who want to travel hassle-free and book their metro tickets without waiting in long queues.

B.6 Conclusion:

We have successfully Designed an interface for an automated ticket vending machine (ATVM) for the local ticket system.