Epicor Mobile Framework

User Interface Design Guidelines
Table of content

- Get Started/Overview
- Guiding Design Principles
- EMF Visual Style Guide
- EMF Components
- EMF User Interface Elements
### Get Started/Overview

Importance of having this guidelines and how it impacts Epicor has brand and overall experience to the users.

<table>
<thead>
<tr>
<th>Importance of these guidelines</th>
<th>Understanding of the Fundamentals</th>
</tr>
</thead>
<tbody>
<tr>
<td>When starting a new application or redesigning an existing one, it is important for developers to restrain themselves from immediately typing code in and start thinking in terms of user experience: what are they trying to achieve, who is the target user of the application, what would the user like, what is the user’s list of desired features, what application use cases should be considered?</td>
<td>UX design for mobile applications is different from desktop ones. Developers need to understand and design within the limits set by several mobile constraints: limited data, limited battery, divided attention, handedness (how devices are held in hand), small screen and network intermittence. The course offers tips on how to design an application taking the constraints into consideration.</td>
</tr>
</tbody>
</table>

### Benefits of knowing and applying these guidelines

Design guidelines are a set of concepts and rules used to create an app’s user interface. These define how the layout on-screen at any one time should be used to maximize the efficiency of presenting data to the user, as well as quickly informing them how to choose their next action.

The main benefits of following design guidelines and using common UI elements in an app development process are summarized as follows:

- User interface consistency, which establishes familiarity and predictability.
- Clarity towards which information is most important and hence visible and usable.
- Reduced user confusion and frustration, leading to improved perception of apps.
- No need to include explicit usage instructions in every app to explain how it must be used.
- Clearer, more efficient and better looking apps.
- Faster development time, improved communication among all the stakeholders involved in developing the interface.
EMF Guiding Design Principles

Ease of Use and Joy of Use are the two critical criteria determining the success of our products and solutions. The best approaches to Successfully & Effectively delivering products that will delight the users is to thoroughly understand the goals of Users and Stakeholders and keep those goals in mind while designing software. It is essential & important for us to Design & Develop products and solutions maintaining rational aspects like efficiency or usefulness with emotional aspects such as friendliness, reliable or trustable.

Our Apps should be: innovative, useful, elegant, understandable, unobtrusive, honest, sustainable, accurate, optimized, minimalist, reliable.

The principles described are critical to the design & development of elegant, efficient and intuitive products.

**Fundamental Principles**

1. **Simplify things to increase confidence of the user**
   Things we create should be easy to use, easy to learn, easy to find, and easy to adapt. Intuition happens outside of conscious reasoning, so by utilizing it you are actually reducing the burden on people’s minds. That will make them feel confident and likely a lot happier.

2. **Time matters, value it**
   Our products should be designed to utilize minimum time of the users to perform/accomplish a task with greater efficiency.

3. **Optimize the design for the most frequent or important tasks**
   Understanding how users will use the software you are designing is critically important. We should use that understanding to anticipate the information, task flows, and features that users require at key points within the user experience.

4. **Predictable/obvious**
   Make your application predictable/obvious by using industry standard user interface conventions wherever possible. Use a common set of design patterns and guidelines so that users don’t have to relearn how to perform common tasks.

5. **Consider persons with disabilities when designing your applications**
   Some users of your application may have impaired vision or physical limitations that affect their ability to use the application.

6. **Contextual help**
   Design the application so that contextual help is available to users when they need it. Users should not have to refer to Help constantly to complete their tasks.

7. **Create a visual hierarchy that matches the user’s needs**
   Visual hierarchy is a combination of several dimensions to aid in the processing of information, such as color, size, position, contrast, shape, proximity to resembling items. Not only must a page be well organized so that it’s easy to scan, but the prioritization of information and functionality ought to imitate real world usage scenarios. Don’t make the most commonly used items the furthest out of reach.

8. **Avoid jargon & technical words**
   Use widely & universally accepted and understood terminology/language only. Speak the users’ language, with words, phrases and concepts familiar to the user, rather than system-oriented terms.

9. **Use appropriate defaults**
   Providing preselected or predetermined options is one of the ways to minimize decisions and increase efficiency. But choose wisely: if you assign the defaults to the wrong options (meaning that the majority of people are forced to change the selection), you’ll end up creating more stress and processing time.
EMF Guiding Design Principles

Fundamental Principles

10. **Provide timely and specific feedback**
   Keep the user informed why they're waiting. Tell them that you're working. Tell them you heard them and offer the next step along their path. User expects feedback when they input a data/information.

11. **Be consistent**
   Navigational mechanisms, organizational structure and metaphors used throughout the design must be predictable and reliable. When things don't match up between multiple areas, the experience can feel disjointed, confusing and uncomfortable. People will start to question whether they're misunderstanding the intended meaning or if they missed a key cue. Consistency implies stability, and people always want to feel like they're in good hands.

12. **Distraction-free experience**
   An effective design should always allow people to focus on the task at hand without having their attention diverted to less critical tasks. Design for tasks to be carried out consecutively instead of concurrently in order to keep people in the moment. The design should be organized with related features and content areas appropriately chunked.

13. **Reduce latency**
   Keep the user informed if they have to wait for processing of particular transaction or data fetching. Use appropriate words to communicate and approximate time they would need to wait.

14. **Error prevention**
   Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. If people make mistakes (either because they misunderstood the directions or mistyped or were misled by you), ensure that they are able to easily fix them.

15. **Be credible and trustworthy**
   It's hard to tell who you can trust these days, so the only way to gain the confidence of your customers is to earn it—do what you say you're going to do, don't over promise and under deliver.

16. **Test the Apps**
   Testing the Apps with actual devices, users and environment and collecting feedback will aid us in improvising the overall experience.
Visual Style Guide

The Epicor Visual Identity is comprised of our logo, design elements, colors, layout style, and image use. The way we present our products and ourselves is second in importance only to the success of the customers we proudly serve. By using this clear and distinctive wordmark, we will maintain a strong presence with all of our audiences in the highly competitive and cluttered environments where we communicate.

Ensuring consistency and legibility wherever our identity is reproduced will amplify the Epicor presence across all media. To that end, it is important to understand, support, and comply with our identity guidelines.


Brand

Epicor Logo usage
The Epicor logo should always have a minimum area of isolation, separating it from content and other distracting elements. This clear zone is effectively the height and width of the Epicor “E”.

Do not reduce the logo to make it fit in the interfaces, always use the largest possible size of height and proportionate width.

Incorrect Logo usage
These examples demonstrate improper use of the Epicor logo. These examples are improper because of illegibility, inconsistency, and lack of regard for Epicor logo guidelines.
Visual Style Guide

Color

The Epicor logo should primarily be represented in blue. It may also be represented in dark gray, or reversed in white with dark gray background.

**Primary Color**

Blue is the used as primary color on all Epicor App interface to keep it in align with the brand and maintain consistency.

**Secondary Color**

Secondary colors are chosen to complement the primary color palette. These colors can be used as headlines and other dominant design elements.

**Note:** To make the color contrast complaint to the Accessibility Guidelines, check the background and foreground and maintain the contrast ratio. Use the online tool to check the http://www.dasplankton.de/ContrastA/.

**Primary Blue**
R18, G195, B244
Hex #12C3F4

**Secondary Dark Blue**
R0, G138, B169
Hex #008aa9

**Secondary Dark Gray**
R134, G137, B141
Hex #86898D
Users should be able to read what they see on the without putting effort into it. In order to achieve it we need to keep the text, which we use to communicate, as readable as possible. Readability is defined as the amount of effort a user puts into reading and understanding text.

Designing for mobile includes the way we present the text on the screen with following aspects.

**Size**

It emphasizes the text and provides the importance which it needs along with the visual hierarchy.

**Whitespace**

We design and develop apps mainly for mobile devices, so we are dealing with smaller screens. To keep providing clarity it is recommended to add space to the text – between lines and in the margins. This is helping users to better interact with the words.

**Contrast**

If text is an important part of the app, texts need to stand out. When defining the contrast of the text and background, keep the high contrast of black and white in mind.

Note: To make the color contrast complaint to the Accessibility Guidelines, check the background and foreground and maintain the contrast ratio. Use the online tool to check the contrast ratio: [http://www.dasplankton.de/ContrastA/](http://www.dasplankton.de/ContrastA/)
Visual Style Guide

Iconography

Icons are a visual representation of an object, action, or idea. If that object, action, or idea is not immediately clear to users, it will make user confusing and frustrating and hinders people from completing a task. Clever and careful use on icons in the App is the key to designing simple and intuitive User Interface screens.

Epicor Mobile Framework uses FontAwesome icon font library (http://fortawesome.io/icons/ or http://fortawesome.io/cheatsheet/) instead of the other form of file type like JPEG, PNG and GIF to leverage advantage like light weight and scalable without distortion. Font EMF also has list of custom icon which are created and updated on need basis. When using icons choose the them wisely so its intuitive, meaningful and can be perceived easily by the users.

FontAwesome Icon sample list

EMF Custom Icon sample list
Visual Style Guide

Animation

When used appropriately, animation can convey status, provide feedback, enhance the sense of direct manipulation, and help users visualize the results of their actions. Animations provide fluid visual transitions between different states of your user interface. Use animations consistently throughout your app.

Animation provides:

• Guided focus between views.
• Hints at what will happen if a user completes a gesture.
• Hierarchical and spatial relationships between elements.
• Distraction from what’s happening behind the scenes (like fetching content or loading the next view).

The most commonly used transitions:

Slide
A slide transition eases a new screen element onto the display by pushing out the prior screen, implying the two screens are side by side.

Cover
A cover transition is modal and is used to present views such as action sheets and on-screen keyboards, which enter and leave via the bottom of the screen. Cover transitions are also used to bring in transient modal elements such as sign-in pages and location-enabling permission requests.

Zoom
Screens are enlarged from the center of the display, irrespective of the location of the tap initiating the transition.
EMF
Components
**EMF Components**

**Shell**

*ep.shell*

**Intro**

*ep.shell* is the outer most container which houses the required UI elements, controls and components. It is the core component necessary to get your app load on the screen, but likely does not contain the visible content.

The shell contains other supporting components like:
- ep.feedback
- ep.menu
- ep.sidebar
- ep.view-container
- ep.views

The shell comprises of the following section:

**Navbar**
Contains navigation buttons and App Title or Brand element.

**Left and Right Nav**
Contains navigation menu, feeds or listings.

**Content Canvas**
Displays the main content or information.

**Footer**
To be used if any optional action buttons or additional info like copyrights.

UI sections like Navbar, Side navbar and Footer can be set to show / hide to make visible on the screen.

[Diagram of shell container]
EMF Components

Shell
ep.shell

Layout appearance

UI sections displayed in mobile and tablet screens. Left and Right Nav will be hidden in the device view, can be accessed either tapping on menu buttons placed on either side of the Nav bar or by swipe gesture.

Mobile view
In mobile view the nav bar, content area footer are displayed when the app loads. Footer can be added for additional functionality or actionable buttons. Side nav menus overlay all other structural elements.

Tablet view
A right nav menu can be accessed temporarily or fixed for permanent display.
EMF Components

Shell
ep.shell.view.container

Intro

ep.shell.view.container is a component used inside any views to control the behaviour, appearance and functions of UI elements like Navbar, Left & Right Nav and Footer. This component gives the provision of controlling the behaviour and appearance of UI element on different device screen size. It also allows to switch buttons appearance and functions in Navbar to navigate between different views or open sidebar nav.

Usage

Use it if
• Use this components to control UI elements in specific views and device screen sizes.

Don’t it if
• If your App has standard UI structure across all views, this component becomes redundant to use.

Behavior and Controls

ep.shell.view.container component allows you the follow settings

sidebarsettings - Settings for Left & Right sidebars - Provide template path.
smallmodesettings - Settings for small screen view (smartphone) - Enable or disable variables.
largemodesettings - Settings for large screen view (tablet) - Enable or disable variables.
btn-home - Settings to button icon and action - Sets the icon for home button and specific action.
EMF Components

Shell
ep.shell.view.container

Responsive and Adaptiveness

With use of `smallmodesettings` and `largemodesettings` control settings behaviour of UI elements can be made responsive to small and large screen devices.

Phone view (Sample screen)
Side nav and footer hidden in small screen mode. 'Value' for both set to 'false'.

Tablet view (Sample screen)
Side nav and footer visible in large screen mode. 'Value' for both set to 'true'. Side nav slides in and visible when tapped of the navbar icon.
EMF Components

Contacts List
ep.contacts.list

Intro

When presenting large number of data in a ListView which are alphabetical in order, similar to contact list in your phone. It will be time consuming to frustrating for user to scroll down all the way to the end to find contact list from letter Z. To make things easier and helpful to the user adding have a alphabetical index scroller to the right side of the screen, allow users to quickly jump to a specific letter without much effort. This component also features a quick search filter which enables user to get the most relevant to their query.

Usage

Use it if
• Use this components when you have list of data to view in alphabetical order.
• Use it when user wants to filter results based on their entry which is huge data list.

Don’t use it if
• If your data list view is combination of alpha numberical entries or contains character symbols.

Behavior and Interaction

ep.contacts.list component features and controls as below:

quick search - Displays the most relevant to user query. If the search query is cleared, the list view gets to the top of the list.
main list view - Allows vertical scroll with swipe gesture in single direction.
alphabetical index - Alphabetical index with ‘#’ for nubmer and A to Z letters, on tap of a letter the list view positions to target listing the entries relevant to that letter.
# EMF Components

## Contacts List

ep.contacts.list

### Appearance

![Diagram of Contacts List with labels:
- List Title
- List Group
- Instant search filter
- Alphabetical Index](image)

<table>
<thead>
<tr>
<th>List Title</th>
<th>List Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Aaric</td>
</tr>
<tr>
<td></td>
<td>Aaron</td>
</tr>
<tr>
<td></td>
<td>Abbot</td>
</tr>
<tr>
<td></td>
<td>Ackerley</td>
</tr>
<tr>
<td></td>
<td>Adam</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alphabetical Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>G</td>
</tr>
<tr>
<td>H</td>
</tr>
<tr>
<td>I</td>
</tr>
<tr>
<td>J</td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>O</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>Q</td>
</tr>
<tr>
<td>R</td>
</tr>
<tr>
<td>S</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>U</td>
</tr>
<tr>
<td>V</td>
</tr>
<tr>
<td>W</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>Y</td>
</tr>
<tr>
<td>Z</td>
</tr>
</tbody>
</table>
EMF Components

Contacts List

Example

Phone view

Tablet view
EMF Components

Icon Selector
ep.icon.selector

Intro
This is a font awesome icon picker which lists all the font awesome icons and their names. This has a text field which displays the selected icon name and a button which displays the icon list on click of it. There is also a search filter inside icon list dropdown which can be used to filter the list.

Usage

Use it if
• Use it when user needs icon as part of any task either to display or associate a meaning or to indicate an action.

Behavior and Interaction

Click on the button on right side of text field. The icon list with names opens up the icon selector drop down. Users can use search box in the drop down to filter the list. The selected icon and name is displayed on the text field.

Following are the attributes which can be passed to this component:

ngModel - (Mandatory) This is the model object to store the selected icon name and it can be used for rest of the application processing.
width - (Optional) This is the width of entire icon selector component. Example values 500px, 80% etc...Default is 100%.
iconListWidth - (Optional) This is the width of icon selector drop down menu. Example values 200px, 300px etc...Default is 295px.
iconListHeight - (Optional) This is the maximum height of icon selector list in the drop down menu. Example values 200px, 300px etc...Default is 400px.
Icon Selector
ep.icon.selector

Appearance

Icon selector

- Icon selector button
- Search filter text box

Icon list with Names

- fa-align-center
- fa-align-justify
- fa-align-left
- fa-align-right
- fa-angle-down
- fa-angle-left
- fa-angle-right
- fa-angle-up
- fa-arrow-down
- fa-arrow-left
- fa-arrow-right
- fa-arrow-up

Icon selector

- On select Icon name is displayed
- On select Icon is displayed
EMF Components

Icon Selector

ep.icon.selector

Example

Phone view

Tablet view
Color Selector

Intro

Color selector component displays the colors in a canvas on click of a button from which the user can choose the desired color.

Usage

Use it if
• Use this components when you want the user to choose a color from the palette that either can an value that should be used or apply it to a particular element.

Behavior and Interaction

On click/tap of the button the color picker field will be displayed and the text field contains the hex format of the color which is selected.

Following are the attributes which can be passed to this component.

• **ngModel**: (Mandatory) This is the model object to store the selected color and it can be used for rest of the application processing.
• **position**: (Optional) This is the position of the color selector dropdown. Allowed values are top, right, bottom, left. Default is bottom.
• **size**: (Optional) This is the size of color selector dropdown in pixels. Example values 200, 300 etc...Default is 100.
EMF Components

Color Selector
ep.color.selector

Appearance

Color selector

- Button to open color selector
- Text field to display hexadecimal color value

Color selector

- Color selector button
- Color slider
- Color field
- Color preview

Selected Hexadecimal color value

#ff6600
EMF Components

Color Selector
ep.color.selector

Example

Phone view

Tablet view
EMF Components

Barcode Scanner

ep.hybrid.barcode

Intro

By using the phone’s built-in camera the scanner will quickly scan and recognize the 2D barcodes, QR Codes and Data Matrix codes and displays the decoded information on the phone screen.

Usage

Use it if

- Use this component when the user wants to get information about a product or a shipment by scanning a barcode or a QR code pattern.

Behavior and Interaction

When you tap on the Scan button it opens up your smartphone camera. By pointing the camera at the Barcode or QR code and lining it up inside the guide arrows recognises it and provides the appropriate information like UPC code and format.

Examples

Barcode

Decoded barcode information
Camera
ep.hybrid.photo

Intro

This component will be used to launch the built-in camera application to take a picture and display it in an ImageView. It will also gain access to the internal photo library and allow user to pick a photo.

Usage

Use it if
• Use this component when you want the user to use the built-in camera of the smartphone or mobile device to take a picture or access existing pictures from the photo library.

Behavior and Interaction

When you tap on the 'Take Picture' button, the smartphone built-in camera application opens up and the user can capture the picture. User can also access the pictures from the photo library of the device.
EMF Components

Camera
ep.hybrid.photo

Example

Option to Take Picture accessing the device camera.

Picture taken accessing device camera

Once the picture is taken the image will be displayed on the screen.

When the user chooses to use picture from device photo library.
EMF Components

Geo Location
ep.hybrid.geolocation

Intro

This component will be used to show map, add marker/multiple markers for current/listed locations. It gives auto complete feature for places, shows driving direction from the source to destination.

Usage

Use it if
• Use this components when you want the user needs to use geographical location features and function with in their work flow or tasks.

Behavior and Interaction

My Location: A marker is placed on the map indicating the current location of the device using the latitude and longitude method.
Pin Address: A marker is placed based on the user input in the text field placed above the map.
Directions: User should enter the ‘Starting point’ and ‘Destination’ to display the best available direction/route on the map.
Autocomplete: When the user starts typing in the text input field and after few characters have been typed into a text input field the system predicts the word a user intends to enter and displays the list to choose from.
Gesture interaction: Zooming and panning.
EMF Components

Geo Location
ep.hybrid.geolocation

Example

**My Location**
Marker placed to indicate the current location.

**Pin Address**
Marker placed on the map based on user input.

**Directions**
Start point and destination marker placed along with the route direction on the map based on user input.
EMF Components

Contacts
ep.hybrid.contacts

Intro

This component will be used to access the contacts list from your device and display the information.

Usage

Use it if
• Use this component if you want the user to use the contacts from their device.

Behavior and Interaction

Tapping the ‘Pick a contact from phone’ opens the device contact list and users can select the desired contact. The selected contact will be displayed on the screen.

Example
EMF Components

Media and Sound
ep.hybrid.media

Intro

This component will be used to play sound on trigger.

Usage

Use it if
• Use this components when the user wants to play an externally linked audio file or when you want the user’s attention by audio notification.

Behavior and Interaction

Tapping the ‘Play music’ plays the audio till the end and can be stopped by tapping ‘Stop’ button.

Example

Play Sounds

- Play Music Button
- Stop Music Button
EMF Components

Vibrate
ep.hybrid.vibration

Intro

This component will be used to vibrate the device on trigger.

Usage

Use it if
• Use this component when want the user's device to vibrate on trigger or seeking user's attention for any action performed.

Behavior and Interaction

Tapping the 'Trigger Vibration' vibrates the device for duration of 3 seconds.

Example

![Trigger vibration Button]
EMF Components

Device Information
ep.hybrid.device

Intro

This component will be used to display the device information like Model, Platform, ID and Operating System version.

Usage

Use it if
• Use this component when you want the user to view information of the device he/she using.

Example

![Device Information](image-url)
EMF Components

Flashlight
ep.hybrid.flashlight

Intro

This component will be used to switch on/off flashlight the of mobile device.

Usage

Use it if
• Use this components when user’s needs to use the flashlight of the mobile device they are using.

Behavior and Interaction

Tapping on the ‘On’ button will switch on the flashlight / torch of the device and ‘Off’ button will switch off the flashlight / torch of the device.

Example

![Flashlight Example Image]
EMF Components

Network Status
ep.hybrid.network

Intro

This component will be used to detect and display the Network Status of the device like internet status and type.

Usage

Use it if
• Use this component when the users want to know the network status of their device.

Example

![Network Status Example](image)

- Internet connection status
- Internet connection type
EMF Components

Calendar
ep.hybrid.calendar

Intro

This component will be used to access and interact with device calendar features and has the ability to create and delete Calendar and Events and set reminders as well. It can also list the available list of calendar on the device.

Usage

Use it if

• Use this components when the users wants to read or write the date and time based information from the mobile device.

Behavior and Interaction

View Calendar: View the default device calendar.
Create Event: Lets the user create an event where you can details like title, location, notes and start-end date and time.
Create Event with options: In addition to create event features, additional options like reminders and recurrence can be set.
Find Event: This will find and fetches the details of an from the device calendar.
Delete Event: Lets the user find specific event and delete it from the device calendar.
List Calendar: Displays the list of all the calendars available in your device.
List Events: Displays the list of all the events available in your device.
EMF Components

Calendar
ep.hybrid.calendar

Examples

Default device calendar

Create Event

Create Event with options like reminder and recurrence
EMF Components

Calendar

ep.hybrid.calendar

Examples

Find Events
Delete Events
List of calendar from the device
List of events from the device calendar
EMF
User Interface Elements
Grid System

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the grid system works:

- Rows must be placed within a .container (fixed-width) or .container-fluid (full-width) for proper alignment and padding.
- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like .row and .col-xs-4 are available for quickly making grid layouts. Less mixins can also be used for more semantic layouts.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three .col-xs-4.

```
.col-md-4
.col-md-4
.col-md-4
.col-md-8
.col-md-4
```
EMF UI Elements

Responsive Utilities

For faster mobile-friendly development, use these utility classes for showing and hiding content by device via media query.

**Available classes**

Use a single or combination of the available classes for toggling content across viewport breakpoints.

<table>
<thead>
<tr>
<th></th>
<th>Extra small devices Phones (&lt;768px)</th>
<th>Small devices Tablets (≥768px)</th>
<th>Medium devices Desktops (≥992px)</th>
<th>Large devices Desktops (≥1200px)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.visible-xs-*</td>
<td><strong>Visible</strong></td>
<td>Hidden</td>
<td>Hidden</td>
<td>Hidden</td>
</tr>
<tr>
<td>.visible-sm-*</td>
<td>Hidden</td>
<td><strong>Visible</strong></td>
<td>Hidden</td>
<td>Hidden</td>
</tr>
<tr>
<td>.visible-md-*</td>
<td>Hidden</td>
<td>Hidden</td>
<td><strong>Visible</strong></td>
<td>Hidden</td>
</tr>
<tr>
<td>.visible-lg-*</td>
<td>Hidden</td>
<td>Hidden</td>
<td>Hidden</td>
<td><strong>Visible</strong></td>
</tr>
<tr>
<td>.hidden-xs</td>
<td>Hidden</td>
<td><strong>Visible</strong></td>
<td>Hidden</td>
<td><strong>Visible</strong></td>
</tr>
<tr>
<td>.hidden-sm</td>
<td><strong>Visible</strong></td>
<td>Hidden</td>
<td><strong>Visible</strong></td>
<td><strong>Visible</strong></td>
</tr>
<tr>
<td>.hidden-md</td>
<td><strong>Visible</strong></td>
<td><strong>Visible</strong></td>
<td>Hidden</td>
<td><strong>Visible</strong></td>
</tr>
<tr>
<td>.hidden-lg</td>
<td><strong>Visible</strong></td>
<td><strong>Visible</strong></td>
<td><strong>Visible</strong></td>
<td>Hidden</td>
</tr>
</tbody>
</table>
EMF UI Elements

Tables

We use tables all the time to represent data sets. Tables are built with columns and rows. Columns have headings and everything under them is associated with those headings. Likewise, rows can have headings so that everything in that row is associated with those headings. With help of predefined classes available in EMF you can create tables with different style depending on the need or purpose of displaying the data.

When displaying information in tables, its good practices to have border, horizontal and vertical divider lines to visually differentiate between each cells.

<table>
<thead>
<tr>
<th>No.</th>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mark</td>
<td>Thornton</td>
<td><a href="mailto:mark@mail.com">mark@mail.com</a></td>
</tr>
<tr>
<td>2</td>
<td>Jacob</td>
<td>Otto</td>
<td><a href="mailto:jacob@mail.com">jacob@mail.com</a></td>
</tr>
<tr>
<td>3</td>
<td>Larry</td>
<td>Winston</td>
<td><a href="mailto:sam@mail.com">sam@mail.com</a></td>
</tr>
<tr>
<td>4</td>
<td>John</td>
<td>Sam</td>
<td><a href="mailto:win@mail.com">win@mail.com</a></td>
</tr>
</tbody>
</table>

Always use the `<th>` tag enclosed inside `<thead>` for the table headings.

To differentiate table rows from each other, a different shade is used as background color for every second row.

Use `.table-striped` to add zebra-stripping to any table row within the `<tbody>`. 
EMF UI Elements

Buttons

There's seven built-in color styles to choose from when working with buttons in your Apps. To use them, simply add .btn btn-option (where btn-option is the style you want to set the button to) to a <button></button> tag eg: <button type="button" class="btn btn-default">Default</button>.

Style Button

Use any of the available button classes to quickly create a styled button.

Available options

- Default: class: btn btn-default
- Primary: class: btn btn-primary
- Success: class: btn btn-success
- Info: class: btn btn-info
- Warning: class: btn btn-warning
- Danger: class: btn btn-danger
- Link: class: btn btn-link

Button Sizes

To control the size of buttons, simply add .btn-lg, .btn-sm or .btn-xs to a button eg: <button type="button" class="btn btn-primary btn-lg">Large button</button>.

Available options

- Large button: class: btn-lg
- Default button: class: btn-primary
- Small button: class: btn-sm
- Extra small button: class: btn-xs
EMF UI Elements

Forms

Forms are integral part of any of the transaction or task oriented Apps. When working with form elements it requires style control methods to make it look and behave consistently across. EMF greatly simplifies the styling process of form controls like inputboxes, selectboxes, textareas etc.

Individual form controls automatically receive some global styling. All textual `<input>`, `<textarea>`, and `<select>` elements with `.form-control` class are set to width: 100%; by default. Wrap labels and controls in `.form-group` for optimum spacing.

Example

```
Email address
```

Labels enclosed in `<label>Email address</label>` tag.

```
name@email.com
```

Placeholder text used as example to hint the user required info.

```
name@email.com
```

Default state

```
name@email.com
```

Focus state

```
name@email.com
```

Disabled state

Add `disabled` attribute to any form element to prevent user interactions
You can use the disable attribute to field set to disable the entire field set.

```
name@email.com
```

Readonly state

Add `readonly` attribute to any form element to prevent modification of the input's value
You can use the disable attribute to field set to disable the entire field set.
EMF UI Elements

Images

Images artifact that depicts visual perception or appearance of physical object or a person. When using images in your apps make sure your images are appropriately sized for displays and across platforms. Images shouldn’t appear pixelated or distorted, so test appropriate resolution sizes for specific ratios and devices.

When using images be consistent in the width and height of similar images. Always use .PNG (Portable Network Graphics) file format and optimize the images for best quality and lowest file size.

Maximum care should be given when using images from web or any other source for copyright violations.

Image styling

Using the frameworks built-in classes you can easily style images such as making the ‘Round Cornered’ or ‘Circular Shape’ or give them effect like ‘Thumbnails’.

Round Cornered
Class: .img-rounded

Circular Shape
Class: .img-circle

Thumbnails
Class: .img-rounded
EMF UI Elements

Typography

EMF uses Helvetica Neue, Helvetica, Arial, and sans-serif in its default font stack. Using typography feature of framework you can create headings, paragraphs, list and other inline elements.

Headings

h1. Bootstrap heading
h2. Bootstrap heading
h3. Bootstrap heading
h4. Bootstrap heading
h5. Bootstrap heading
h6. Bootstrap heading

Body copy

Bootstrap's global default font-size is 14px, with a line-height of 1.428. This is applied to the <body> and all paragraphs. In addition, <p> (paragraphs) receive a bottom margin of half their computed line-height (10px by default).

Inline text elements

Marked text
For highlighting a run of text due to its relevance in another context, use the <mark> tag.

You can use the mark tag to highlight text.

Underlined text
To underline text use the <u> tag.

This line of text will render as underlined

Bold
For emphasizing a snippet of text with a heavier font-weight use <strong> tag.

The following snippet of text is rendered as bold text.
EMF UI Elements

Helper / Utility Classes

EMF includes variety of utilities classes with a single purpose. They’re designed to reduce the frequency of highly repetitive declarations in your CSS while allowing for quick and easy development.

Contextual text and background colors
Convey meaning through color with a handful of emphasis utility classes.

<table>
<thead>
<tr>
<th>Text</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example text</td>
<td>class=&quot;text-muted&quot;</td>
</tr>
<tr>
<td>Example text</td>
<td>class=&quot;text-primary&quot;</td>
</tr>
<tr>
<td>Example text</td>
<td>class=&quot;text-success&quot;</td>
</tr>
<tr>
<td>Example text</td>
<td>class=&quot;text-info&quot;</td>
</tr>
<tr>
<td>Example text</td>
<td>class=&quot;text-warning&quot;</td>
</tr>
<tr>
<td>Example text</td>
<td>class=&quot;text-danger&quot;</td>
</tr>
</tbody>
</table>

Close icon
Use the generic close icon for dismissing content like modals and alerts.

×       class="close"

Carets
Use carets to indicate dropdown functionality and direction. Note that the default caret will reverse automatically in dropup menus.

▼     class="caret"

Quick floats
Float an element to the left or right with a class.

class="pull-left"       class="pull-right"
EMF UI Elements

EMF Custom Utility Classes

Apart from standard framework CSS helper classes EMF has a set of customized classes to make UI development hassle free. Custom classes are created by EMF team based on the development needs and challenges. Class name which starts with .ep are all custom classes available.

Alignment Classes
Similar to quick float classes additional custom classes are made available for positioning the UI elements in the screens.

**Horizontal Alignment**
- .ep-align-hleft - Aligns the div to horizontal left.
- .ep-align-hcenter - Aligns the div to horizontal center.
- .ep-align-hright - Aligns the div to the horizontal right.

**Vertical Alignment**
- .ep-align-vtop - Aligns the div to vertically top.
- .ep-align-vcenter - Aligns the div to vertically center.
- .ep-align-vbottom - Aligns the div to the vertically bottom.

**Combining Classes for Multi Axis Alignment**
To position the elements horizontally as well as vertically you can combine any of horizontal and vertical classes, like .ep-align-hright.ep-align-vbottom.
EMF UI Elements

EMF Custom Utility Classes

Padding & Margin
To create space inside and outside of an element with these custom classes. For specific needs separate classes with different properties like left, right, top & bottom and values like 0 5 10 15 & 20 are available. For margin negative value classes are available as well to place the elements.

Padding
.ep-pad-all-0              - 0 padding on all sides
.ep-pad-top-0             - 0 padding on top
.ep-pad-right-0           - 0 padding on right
.ep-pad-bottom-0          - 0 padding at bottom
.ep-pad-left-0            - 0 padding on left
.ep-pad-vert-0            - 0 padding on top and bottom
.ep-pad-horz-0            - 0 padding on left and right

Value 0 is replaced by 5, 10, 15 and 20

Margin
.ep-margin-all-0            - 0 margin on all sides
.ep-margin-top-0            - 0 margin on top
.ep-margin-right-0          - 0 margin on right
.ep-margin-bottom-0         - 0 margin at bottom
.ep-margin-left-0           - 0 margin on left
.ep-margin-vert-0           - 0 margin on top and bottom
.ep-margin-horz-0           - 0 margin on left and right

Value 0 is replaced by 5, 10, 15 and 20
EMF UI Elements

Navbar

The navbar is a simple wrapper for positioning branding, navigation, and other elements into a concise navigation header.

Brand image

Replace the navbar brand with your own image by swapping the text for an `<img>`. Since the `.navbar-brand` has its own padding and height, you may need to override some CSS depending on your image.

Forms

Place form content within `.navbar-form` for proper vertical alignment and collapsed behavior in narrow viewports. Use the alignment options to decide where it resides within the navbar content.

Buttons

Add the `.navbar-btn` class to `<button>` elements not residing in a `<form>` to vertically center them in the navbar.

Text

Wrap strings of text in an element with `.navbar-text`, usually on a `<p>` tag for proper leading and color.

Non-nav links

For folks using standard links that are not within the regular navbar navigation component, use the `.navbar-link` class to add the proper colors for the default and inverse navbar options.
EMF UI Elements

Pagination & Pager

Pagination control should be provided to browse from screen to screen. Helps the user browse to the previous and next pages by providing links to such actions. Also, provide links to the absolute start and end of the dataset (first and last).

Default pagination
Links are customizable for different circumstances. Use .disabled for unclickable links and .active to indicate the current page.

Pager
Sometimes you may simply require previous and next links to navigate between few sets of screens, in such cases you are use the pager class to style the links.
EMF UI Elements

Progress Bars

A progress bar can be used to show that assets are loading, in progress or that there is action taking place. Progress bar helps the user understand or know that status of an action performed.

Default progress bar

With label

To ensure that the label text remains legible even for low percentages, add a min-width to the progress bar.

Contextual alternatives

Progress bars use some of the same button and alert classes for consistent styles. You can use the classes progress-bar-success, progress-bar-info, progress-bar-warning or progress-bar-danger.

Stacked

Place multiple bars into the same .progress to stack them.
EMF UI Elements

Modal popup

Modal are basically a dialog box that is used to provide important information to the user or prompt user to take necessary actions before proceeding further.

Title

Cras mattis consectetur purus sit amet fermentum. Cras justo odio, dapibus ac facilisis in, egestas eget quam.

Button

Action button

Close button

Transparent overlay mask
EMF UI Elements

Tab & Pill

EMF provides styling classes for navigation components like Tabs and Pills. `.nav` is required as the base class in the markup to get the styling.

Tabs

`.nav-tabs` class is used to get the tab style and `.active` is applied to indicate the active tab.

Settings  Profile

Pills

`.nav-pills` class is used to get the pills style and `.active` is applied to indicate the active tab.

Settings  Profile

Combining `.nav-pills` class along with `.nav-stacked` we can vertically stack the links.

Home  Settings  Profile
EMF UI Elements

Tooltip & Popover

Tooltip
Tooltip are simple text labels that appear when the user hovers over, focuses on, or touches an element to indicate a hint or tip to the users on what that element is meant.

By default, tooltip is set to on top of the tapped element by the tooltip plugin. You can set the tooltips to appear on top, right, bottom and left sides of an element by using the data-placement attribute and then choose your direction.

Popover
Any time you need to have more actions available to the user than the confined space allows, Popover are used to display those information or actionable elements. Popover allows to display title and body content.

Similar to tooltip by default, popover is set to on top of the tapped element by the popover plugin. You can set the popovers to appear on top, right, bottom and left sides of an element by using the data-placement attribute and then choose your direction.
EMF UI Elements

Carousel

A slideshow component for cycling through element like images or slides of text like a carousel. Cyclic feature can be controlled and it can either be continous or hard stop.

The slideshow can be controlled through data-attribute or javascript with different options.

- **interval** - The amount of time to delay between automatically cycling an item. If false, carousel will not automatically cycle.
- **pause** - If set to "hover", pauses the cycling of the carousel on mouseenter and resumes the cycling of the carousel on mouseleave.
- **direction** - The direction in which the carousel should slide either "left" or "right".
**EMF UI Elements**

**Collapse**

The collapse plugin makes it easy to make collapsing and expanding a block of content in-line. Showing only the most essential features/content up front, and hide the rest. Reveal secondary elements “inline” when requested by the user.

On tapping the button or a link the show/hide features works by dynamically swapping the classes as per the current state.

- `.collapse` - Hides content
- `.collapsing` - Applied during transitions
- `.collapse.in` - Shows content

**Default collapse example**

Show part details

enim eiusmod high life accusamus terry richardson ad squid. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident.

Expanded content block

**Accordion example**

The default collapse behavior can be extended to create an accordion with the panel component.

- Show part details

enim eiusmod high life accusamus terry richardson ad squid. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident.

- Show billing address

- Show customer details

Expanded panel

Collapsed panel
EMF UI Elements

Alert

Provide contextual feedback messages for typical user actions with the handful of available and flexible alert messages.

For clear communication and proper styling, use one of the four required contextual classes.

- .alert-success
- .alert-info
- .alert-warning
- .alert-danger

Simple alert example

| Well done! You successfully read this important alert message. |
| Heads up! This alert needs your attention, but it's not very important. |
| Warning! Better check the message, before it's too late. |
| Oh snap! Change a few things up and try submitting again. |

Dismissible alert example

User has to control to dismiss or close the alert box by tap on the close button.
EPICOR MOBILE FRAMEWORK

Core Team - Please refer the link to know about the team behind EMF http://epicoremf.azurewebsites.net/#/team

EMF Maven / Contact (Distribution list) - DL-EMF-DISCUSSION

Version - Read release notes for all the amendments and modifications from the previous version, refer http://epicoremf.azurewebsites.net/#/emfreleases

Developer Help - Anyone who wants to understand or know more refer http://epicoremfkb.azurewebsites.net/