

## A Novel Approach for Designing Mobile Native Apps

Sasmita Pani\*, Jibitesh Mishra

College of Engineering and Technology, Bhubaneswar, India

### ARTICLE INFO

Article history:

Received: 07 December, 2016

Accepted: 14 January, 2017

Online: 28 January, 2017

Keywords:

Content Design

Architecture Design

Conceptual Design

Personification Design

### ABSTRACT

Mobile devices are differed from desktop based systems in terms of particular execution environment, constrained resources, and high mobility requirement. To overcome these shortcomings, various agile based methodologies are developed for native mobile applications such as Mobile-D, Scrum etc. These agile techniques are based on various phases and these phases begin from exploring, initializing and implementing the mobile apps. But these techniques are not focusing on elaborating design model for mobile native apps. The aim of the paper is to provide a layered approach or layered model for design of mobile native apps which can be used as a framework for developing mobile native apps. Any mobile native app developer can use this sequential approach or design model for design and development of mobile native apps. This design model gives a standard or framework, based on which generic native mobile apps can be designed and developed. This paper also shows an empirical analysis among the web app design models with the proposed design model for mobile native app development.

## 1. Introduction

Mobile devices are the pervasive based systems embedded with various sensors and powerful processors which can provide information about any domain like agriculture, health care system and learning system. There are various platforms of mobile phones like ios phone, windows phone, android phone where mobile native apps are developed. The agile software methodology is used to develop any mobile apps but this approach is not taken consideration about design modeling of generic native apps under mobile domain. Generally the agile process models are using HTML, HTML5, JavaScript, XML and CSS for design of mobile native apps. These agile process models doesn't provide design model for mobile native apps. There also exists various web app design models such as UWE, WebE, OOHDM and OOWS for designing web apps but not for mobile native apps. We have developed a design model consisting of seven design phases such as content design, architecture design, conceptual design, user interface design, navigation design, presentation design and personification design. The content design describes about how the contents of a generic mobile native app will be organized, constructed and visualized on the screen of a mobile device. The architecture design focuses the specific way of describing the various components needed for developing generic mobile native

apps. The conceptual design specifies about how the generic mobile native app will build semantically and consistently so that user can get meaningful information from the mobile apps. The user interface design identifies various widgets, views, input control elements, layouts etc. which enables the user interactions with the native mobile apps in a ubiquitous manner. The navigation design describes about how a user will navigate from one screen to another screen in a native mobile apps. The presentation design focuses on various transition and animation mechanisms for efficient visualization of mobile native apps. The personification design discusses about integration of social apps with the mobile native apps so that user can get personalized information and can be benefited.

Here the research process is to identify and provide a design model for generic mobile native apps which incorporates various design elements like content, architecture, user interface, transition animations and social media tools etc and provide an empirical analysis among the web app design models with the proposed mobile native app design models.

The rest of the paper is organized as follows. Section 2 discusses related works on web app development process models. Section 3 provides the research approach for developing design model for mobile native apps. Section 4 discusses design elements and proposes a design model for mobiles native apps. Section 4 also gives an empirical study about the existing web app models

\*Corresponding Author: Sasmita Pani, College of Engineering and Technology, Bhubaneswar, India, +919776216142, susmitapani@gmail.com

and the proposed mobile native app design model. This section also discusses the different design phases for the development of mobile native apps. Section 5 provides the development practices of different design phases in mobile native apps. Section 6 provides conclusion and future work respectively.

## 2. Background Study

### 2.1. Web App Design models or Process Models

WebE [1], is a web development methodology employs agile process model when the web app development process is considered as an immediacy and continuous evolution process. This methodology follows incremental process model when the web app development process is developed over a long period of time. The WebE process framework consists of various phases such as customer communication, planning, modeling, construction, delivery and evaluation. Customer communication phase includes business analysis and formulation as steps to build a web app. Business analysis determines the business or organizational context for the web app. The business analysis includes motivation, usefulness and specifies users of the web app to be developed. The formulation phase involves requirement gathering activity which includes all stakeholders for describing the problem that the web app is to be solved. The planning phase consists of task definition and a timeline schedule for the time period used for the development of web app. The modeling phase includes analysis model and design model. The analysis model provides a basis for web app design. It analyses contents, user's interaction, usage scenarios and a particular environment for a web app to be developed. The design model employs various design activities such as content design, aesthetic design, architecture design, interface design, navigation design and component design.

The object oriented hyper design method (OOHDM) [2], includes hypermedia applications in four phases i.e. conceptual design, navigational design, abstract interface design and implementation. These phases are performed in a mix of incremental, iterative and prototype-based development styles. During each activity a set of object-oriented models are given to describe about how a particular design concern is built or enriched from previous iterations. In the conceptual phase, a class scheme is built out of sub-systems, classes and relationships. Navigational design is considered a critical step in the design of a hypermedia application. Each navigational model is built as a view over conceptual models. It is expressed into two schemas the navigational class schema and navigational context schema. OOHDM provides ADV (Abstract Design View) approach provides the way in which different navigational objects will look and which interface objects will activate the navigation. The abstract interface design activity specifies interface objects at abstract level through which the user will perceive. In this implementation phase, the designer will actually implement the design. In this phase, the designer has to decide how the conceptual and navigational objects will be stored. It can be done by mapping of object oriented model to relational model.

WebML [3], is modeling language for designing web sites at the conceptual level. WebML supports XML syntax which enables the high-level description of a web site. WebML consist of four orthogonal perspectives or aspects for specifying a web

site. These are structural model, hypertext model, presentation model and personalization model. From the structural design point of view, it undergoes in two phases such as requirement gathering and data modeling phases. The requirement gathering phase includes objective of the web site, style guideline of the web content, constraints due to legacy data. The hypertext design aspect considers two ways of hypertext design models such as composition model and navigational model. The composition model also can be defined as by designing the "hypertext at large" since it specifies pages, units and provides hyperlink among the pages through units. The composition model specifies nodes for making up the hypertext contained in the web site. This model also defines the pages that compose the hypertext and the content units those make up a page. Six types of content units can be used to compose pages: data, multi-data, index, filter, scroller and direct units. The presentation model specifies the layout and graphic appearance of pages, independently of the output device and of the rendition language that is used generating web pages. It adds to each page a presentation style after all pages are sufficiently stable. It is concerned with the actual look and feel of the pages identified by composition modeling. From the personalization perspective, it enables two ways of design approach for designing web which includes user and group design. In the user and group modeling, user and groups are explicitly modeled in the structure schema.

UWE [4], stands for UML based web engineering, is composed of different models at different phases such as requirements engineering, analysis, design, and implementation of the development process which are used to represent different views of the web application that includes content view, navigation view and presentation view. UWE includes UML diagrams to represent structural aspect of different views and behavioural aspect of web apps. Requirement modeling is done at two steps. At first a rough description of the functionalities is produced by UML use case diagram. In this step, UWE provides three types of use cases such as navigation, process, and personalized use cases. Navigation use cases are used to model typical user behaviour when interacting with a web application. Process use cases are used to describe business tasks that end users will perform with the web apps. Personalized use cases are used to provide customization services to user with in web apps. In a second step, a more detailed description of the functionalities is developed by UML activity diagrams that give the responsibilities and actions of the stakeholders. The navigation model comprises the navigation space model and the navigation structure model. The first specifies which objects can be visited by navigation through the web application and the second specifies how these objects are reached. The navigational space model specifies two modeling elements such as navigation class and navigation association. These elements are termed as page and link in web terminology. The presentation model, is based on the navigation model provides an abstract view of the user interface of a web app. The presentation model only describes the basic structure of the user interface by including UI elements such as text, images, anchors, forms. These UI elements are used to present the navigation nodes. The presentation model also

specifies how each node or page is presented to a user and how the user can interact with them.

OOWS is an object oriented web solution method [5], which provides a strategy for going from the problem space, is presented as conceptual models to the solution space that is the final software product in an automatic way. The OOWS web app development methodology is composed of two main phases such as conceptual modeling and solution development. The conceptual modeling phase is again decomposed into three sub phases sub steps such as functional requirements elicitation, classic conceptual modeling, navigation and presentation modeling. In the functional requirement elicitation phase, conceptual schema is built by applying use cases and scenarios of university department by taking it as case study. In the classic conceptual modeling sub phase, structural model, functional and dynamic models are used to capture the system structure and behaviour. The navigational modeling is done in two phases which includes user identification and categorization step and navigational diagram specification step. This user identification and categorization step provides a user diagram which specifies the system kind of users and the interaction among users. The diagram also specifies the accessibility of each kind of user to the system information and functionality. The navigational diagram specification phase provides a navigation model to capture the navigational requirements of web apps by defining a navigational view or navigational map for each kind of relevant users of the system. The presentation modeling provides various presentation patterns such as information paging pattern, circularity pattern and layout patterns which are used to specify navigation features and capture the essential requirements for the web app development. The specified information will be used by the model compiler to generate the web pages interface. The solution development is the second main step in the OOWS approach which is determined by the target platform and is chosen a specific architectural style.

WSDM [6], stands for web site design method, is an audience driven methodology which does the requirement analysis from audience point of view unlike the other web apps development methodologies like OOHDM, WebML, OOWS, UWE and WebE. The WSDM is a user centric web app development approach and it integrates with V-model for testing of each phase to enhance the maintainability and effectiveness of the web site. WSDM phases include mission statement phase, audience modeling phase, conceptual design phase, implementation design phase and implantation phase. The mission statement phase defines the subject, purpose and specifies the intended users and declares them as target audience. It ensures that the designer should clearly establish the borders of the design. The audience modeling phase provides users a general indication of the audiences involved in the web app. The users identified in the mission statement are taken as a starting point and classified into different audience classes based on their information and functional requirements.

### 3. Research Approach

In order to explore the issues around mobile native app design specification processes and what characteristics are typically

included in these design specifications, we have established three research questions (RQ1to RQ3). These are as follows.

RQ1. How the design phases are developed from various participants for developing mobile native apps?

This question RQ1 was established to develop the design phases through sending questionnaires to different mobile app developers of different software industries. Based on the responses we have obtained design elements and design phases for mobile native app development.

RQ2. How the design models are defined and modeled in mobile native apps considered in existing process models under web app domain for effective design?

In order to define the design models and modeling them in developing mobile native apps, RQ2 was established to examine the current process trends into designing web applications through empirical study, with a particular focus developing the design phases in mobile native apps through mapping the current web app design models and its attributes into those of mobile native apps.

RQ3. How the different attributes or elements under mobile native app development are identified in order to give effective design for mobile native apps.

For specifying the design elements and modeling them in semantic way, RQ3 was established to identify and understand the attributes or elements that lean in different design phases using different specifications under mobile domain.

We undertook an extensive set of questionnaires (60) based on these research questions and send these questionnaires to different mobile app software developers of different software industries of India having international offices as well. The questions were placed in Google forms and sent to different mobile app developers of different software industries. After getting their responses, we are able to find out different attributes or elements used in different design phases under mobile native app domain. All these sending questionnaires and response processes were conducted over a period of 6 weeks, primarily during October/November 2015.

## 4. Proposed Design Model for Mobile Native Apps

Although different mobile app development methodologies describes various phases such as preparation, design, code generation, testing and commercialization of the product for developing mobile apps but they don't emphasize on design modeling of mobile native apps. There is no specific design model for mobile native generic apps so that any user can design the generic mobile native apps and based on the design model, development for mobile native apps can be done. Here our purpose is to develop a design model for mobile native apps which is built by going through the best practices in various software industries.

Some of the questions and responses obtained from the mobile app developers are given below.

### 4.1. Questionnaires through Google Forms

How are the contents organized for designing mobile native apps?

Ans. At first pages are created and XML layouts are used.  
 No of respondents for this answer specified above = 7  
 No of respondents not for this answer=3

Is the sensing mechanism used in architecture design or other design?

- a. Yes
- b. No

No of respondents for a = 8  
 No of respondents for b =2

Which design phase is used for moving one screen to another screen?

Ans. Navigation design  
 No of respondents for this answer specified above = 9  
 No of respondents not for this answer=1

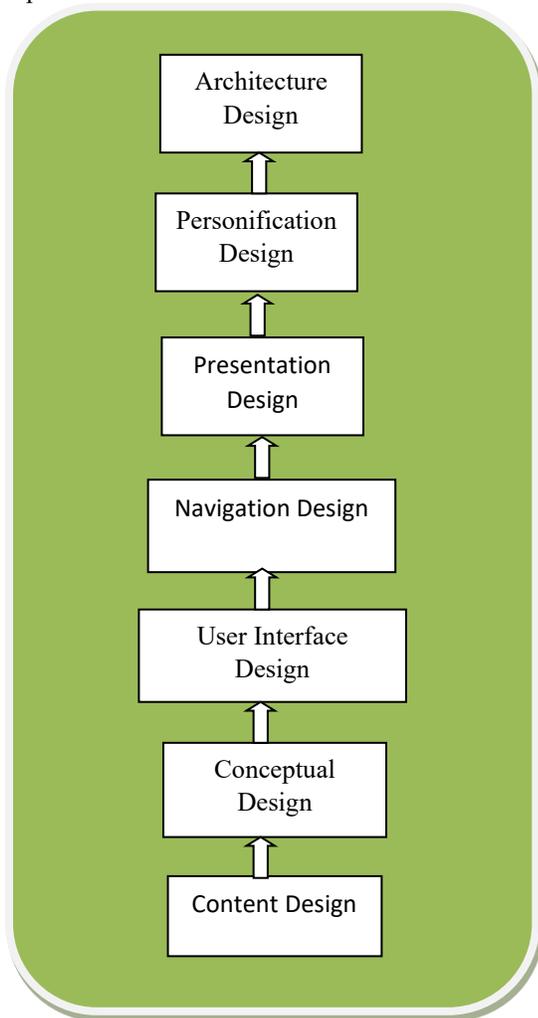


Figure 1. Proposed design model for mobile native apps

After collection of data from different software industries and through the response of different mobile app developers, the design model is built which is consisting of various phases such as content design, architecture design, structural design, user interface design, navigation design, presentation design and personification design for mobile native apps which is shown in fig. 1. Also based on the responses from the mobile app developers,

we have found out various design elements or attributes. Some of the design attributes are shown below.

- a. Different sensors
- b. Keyboard
- c. WLAN
- d. User's role and profile
- e. User's perceiveness

#### 4.2. An Empirical Analysis among the other Web App Design Model with the Proposed Mobile Native App Model

We have studied the existing design models for web app development such as WebE, UWE, and WebML etc. and identified the design phases of those models as discussed in section related work. These design models are used only for designing and developing web apps, but not for mobile native apps. Thus we have proposed and build a design model for mobile native app development which is consisting of seven phases and we have done an empirical analysis among the web app development design models with our proposed design models.

	Design Phases						
	Content Design	Conceptual Design	User Interface Design	Navigation Design	Presentation Design	Personification Design	Architecture Design
<b>UWE</b>	No	Yes	It is included in the presentation model	Yes	Yes	No	No
<b>WebML</b>	No	It is included in the structural model	It is included in the composition model	Yes	Yes	Yes	No
<b>OUIDM</b>	No	Yes	It is included in abstract interface design	Yes	Yes	No	No
<b>WSDM</b>	No	Yes	It is included in the presentation design sub phase of implementation design phase.	It is included in the navigation modeling sub phase of conceptual design phase.	It is included in the presentation modeling sub phase of implementation design phase.	No	No
<b>OOWS</b>	No	It is included in the classic conceptual modeling phase	No	It is included in the conceptual modeling phase	It is also included in the conceptual modeling phase	No	It is included in the solution development phase.
<b>WebE</b>	Yes	It is included in the architecture design phase	Yes	Yes	Yes	No	No
<b>Proposed mobile app design model</b>	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1. An empirical analysis among the web app design model with the proposed mobile native app design model

In the above table 1 provides an empirical analysis of the design phases between the proposed mobile native app design model and the web app design model.

The content design phase is present only in UWE and WebE design model and our proposed mobile app design model, which defines the content to be designed. This design phase is not present in any other web app design models such as OOHDM, WebML, OOWS and WSDM etc.

The architecture design is given only in the OOWS web app design model and our proposed mobile app design model. The architecture design is given in the solution development phase of OOWS web app design model. The conceptual model is provided in the all web app design models as mentioned above in table 1 and our proposed mobile app design model. The architecture design phase in WebE, defines the template which provides consistent hypermedia structure for web app development and hence it is included in the structural model. In OOWS, the conceptual model is included in the classic conceptual modelling sub phase of conceptual modelling phase. The user interface design is present in all the web app design models except OOWS and our proposed mobile app design model. In UWE and WebML, it is included in the presentation model and composition model. In OOHDM and WSDM, it is included in the abstract interface design and presentation design sub phase of implementation phase. The navigation design is also present in all the web app design models and our proposed mobile app design model. In WSDM, it is provided in the navigation modelling sub phase of conceptual design phase. In OOWS, it is included in the conceptual modelling phase. The presentation design is present in all the web app design models and our proposed mobile app design model. In WSDM, it is included in the presentation modelling sub phase of implementation design phase. In OOWS, it is included in the conceptual modelling phase. The personification design is present only in WebML and our proposed mobile app design model.

By far the most significant observation from the responses of questionnaires sending to the mobile native app software developers of different software industries is that identifying the “design phases” or “design aspects” for developing mobile native generic apps. Further these design phases are built in XML language and android studio framework for developing mobile native apps.

4.3. Identifying attributes leaning different design phases in mobile native apps

The device specific attributes includes different sensors, memory, keyboard, gestures and screen resolution. The attributes sensors, memory and keyboards are configured in the system software irrespective of platform and hence it reflects in the architecture design for the generic mobile native app development. The gestures and screen resolution attributes enhances the usability of mobile native apps for the end-user and these attributes are concerned in developing the interface for user in mobile generic native apps. The mobility specific attributes include WLAN, GPRS, EDGE etc and these are configured in the hardware of any platform and hence these are considered in the architecture design of mobile native generic apps.

The user specific attributes include user’s role, profile, user’s perceiveness and user’s usefulness. All these attributes are considered in developing user interface for mobile native apps and hence these attributes reflects in the user interface design of mobile native generic apps. The contents are the views or pages which are designed depending upon all the user specifications and hence all these user specific attributes are concerned in developing content of mobile native generic apps. All the user’s specification attributes are considered in developing conceptual design because user specification is the main aspect for building mobile native generic apps. While developing the navigation, it concerns about user’s perceived ease of use and ease of usefulness towards using the native application in a mobile device. Hence these two attributes such as user’s perceiveness and user’s usefulness are reflected in the navigation design for developing mobile native apps. The social specific attributes are used for improving the social interactions of user’s using the mobile native apps. Hence these attributes are used for optimizing the social tools providers to the users to enhance personification in the mobile native apps.

<i>Different specification under Mobile domain</i>	<i>Attribute’s name</i>	<i>Design phases</i>
<b>Device specific</b>	Different Sensors	Architecture design
	Memory	Architecture design
	Keyboard	Architecture design
	Gestures	User Interface design
	Screen resolution	User Interface design
<b>Mobility Specific</b>	WLAN	Architecture design
	GPRS	Architecture design
	EDGE	Architecture design
	Bluetooth	Architecture design
	GPS	Architecture design
<b>User specific</b>	User’s role and profile	User Interface design

<b>Social specific</b>	User's perceivness	User Interface design
	User's usefulness	User Interface design
	User's perceivness	Navigation design
	User's usefulness	Navigation design
	User's role and profile	Content design
	User's perceivness	Content design
	User's usefulness	Content design
	User's role and profile	Conceptual design
	User's perceivness	Conceptual design
	User's usefulness	Conceptual design
	Blogs	Personification design
	Content Hosting	Personification design
	Social Networking	Personification design
	Podcasting	Personification design
	Wikis	Personification design

Table 2. Attributes in Developing Design Phases of Design Model

#### 4.4. Content Design

Content design plays an important role in developing mobile native apps. It is because the content need to appear on which screen of mobile native apps and hence it need to be organized and constructed very well. At first the contents for generic mobile native apps are organized in each page and each page corresponds to each xml or java script or html file and java activity file depending upon the different platforms of mobile native apps. In android platform, each page corresponds to XML and java activity file which are shown in fig.2.

#### 4.5. Conceptual Design

The agile software methodology is used to develop any mobile apps but this approach is not taken consideration about

semantic modelling of data under the generic mobile domain. Here the approach is to build mobile app in a systematic and semantic way so that the native mobile app user can get the required meaningful information. The conceptual model are built or designed with ontology in the mobile native apps domain to describe the concepts and axioms that determine the semantics of the concepts in that domain.

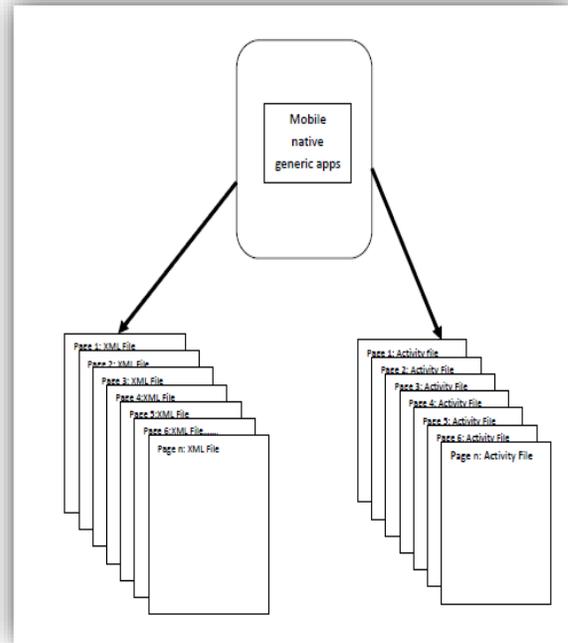


Figure 2. Content Design in Mobile Native Apps

To define the meaning of representing such model to be applied in the semantic web, few mark up languages are specifically available. Web ontology language or OWL [7], is the modern modelling language available to create and represent the ontology having classes and relationships between the classes, clearly describes its meaning so that mobile native application can be developed in a structured, semantic and consistent manner. The conceptual design will work as hidden layer in the mobile native apps so that native app user can get meaningful information. Here we have taken the user specific element or attribute and it has three sub attributes such as based on role, based on preferences and based on usefulness for building mobile native generic app. We have given focus to the user aspect element for designing it semantically because it plays an important role on development of mobile native generic app design model. We have modelled this user aspect element using OWL DL and established class axioms which are shown below.

#### 4.5.1 Building Class Axioms through Making Subclass

For example the user specific element is represented as class and the attributes are represented as subclasses in ontology which is referred as sub class. Here it is done in OWL modelling

language in protégé 5.0 beta framework [8], which is shown below.

```
<!--
http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#BasedOnRole -->
```

```
<owl:Class
rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#BasedOnRole">
  <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#UserSpecific"/>
  </owl:Class>
```

Similarly the attribute based on preference can be made subclass to the class user specific in the ontology for building the semantics.

```
<!--
http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#BasedOnPreferences -->
```

```
<owl:Class
rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#BasedOnPreferences">
  <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#UserSpecific"/>
  </owl:Class>
```

#### 4.5.2 Building Class Axioms through Making Classes Disjoint

To build the ontology consistently and semantically, an instance of one class cannot be instance of another class and hence it is achieved by structuring them as disjoint classes which is modelled through OWL DL language.

```
<rdf:Description>
  <rdf:type
rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#BasedOnPreferences"/>
    <rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#BasedOnRole"/>
    <rdf:Description
rdf:about="http://www.semanticweb.org/ontologies/2015/6/untitled-ontology-176#BasedOnUsefulness"/>
  </owl:members>
</rdf:Description>
```

#### 4.6 User Interface Design

The user interface design takes into account various elements for designing mobile native apps. The elements are specific to platform for a mobile generic native app development. In android platform [9], the elements for designing user interface are in mobile native are

- Different layouts
  - Linear layouts
  - Relative layouts
  - Frame layouts
  - Table layout
  - Grid layout
- Widgets
  - Buttons

- Radio buttons
- Toggle buttons
- Image buttons
- Text views
- Image views
- Web views
- Progress bars
- Check box
- Rating bars
- Date and time pickers
- Menu

- Containers
  - Scroll view
  - Grid view
  - Radio group
  - List view

#### 4.6.1 Designing Layouts

The layouts for designing mobile native apps should be matched or wrapped with screen size of the mobile device which is referred as `match_parent` or `wrap_content` as shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
</RelativeLayout>
```

#### 4.6.2 Designing Widgets

##### 4.6.2.1 Designing Textviews

To view text we can use the tag `<TextView>` and to provide input during run time we can use the tag `<EditText>` for designing mobile native apps. The attributes of the textviews are height, width, id and text etc. The height and width of the text view and edit text can either be matched with the parent layout or enclosed around the content of the text.

```
<TextView
android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Hello World!"
  android:id="@+id/textView" />[Here id identifies
the textView anywhere in the code]
<EditText
android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="Hello World!"
  android:id="@+id/EditText" />
```

Here if more than one textview and edittexts are added, then they should be labeled according to their occurrence in the mobile native apps.

#### 4.6.2.3 Designing Buttons

There exists various buttons such as general buttons, toggle buttons, radio buttons and image buttons for designing mobile native apps. The buttons are used for initializing an event or action corresponding to the respective stimuli or input. The attributes defined for buttons are height, width, id, source and text. The attribute "text" specifies text displayed on the button. The attributes are defined within the specified button tag like <Button> and they are represented as follows.

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Press Button"
    android:id="@+id/button"
/>
<RadioButton
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="Yes"
    android:id="@+id/radioButton"
/>
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/imageButton"
    android:src="@drawable/thumb1small" [src :
attribute is used for accessing the image on the
button]
```

#### 4.6.3 Designing Containers

The containers are used to present group of elements in a single view or page. The elements like group of radio buttons, checkboxes, list of items, textviews etc. are represented through containers. The different types of containers are scrollview, listview and radiogroup etc. and the attributes specified same as above with some new attributes such as discussed below.

```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/andro
oid"
    android:id="@+id/content"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView style="?android:textAppearanceMedium"
android:lineSpacingMultiplier="1.2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/lorem_ipsum"
        android:padding="16dp" />
</ScrollView>
```

Here <ScrollView> is the container tag with in which any widgets or group of widgets can be defined and it reduces the code complexity by grouping the like elements together with in container.

Note: - For every activity\_XML file, there is a corresponding main activity java file or individual java activity file which contains all the required packages, methods, references and instances to perform the various actions and the corresponding listener methods irrespective to mobile native apps.

The containers are used to present group of elements in a single view or page. The elements like group of radio buttons, checkboxes, list of items, textviews etc. are represented through containers. The different types of containers are scrollview, listview and radiogroup etc. and the attributes specified same as above with some new attributes such as discussed below.

```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/andro
oid"
    android:id="@+id/content"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView style="?android:textAppearanceMedium"
android:lineSpacingMultiplier="1.2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/lorem_ipsum"
        android:padding="16dp" />
</ScrollView>
```

Here <ScrollView> is the container tag with in which any widgets or group of widgets can be defined and it reduces the code complexity by grouping the like elements together with in container.

Note: - For every activity\_XML file, there is a corresponding main activity java file or individual java activity file which contains all the required packages, methods, references and instances to perform the various actions and the corresponding listener methods irrespective to mobile native apps.

#### 4.7 Navigation Design

The navigation mechanism provides traversing between the different screens or pages or views with in a mobile native app. The navigation is performed in various ways such as top down navigation, front navigation, back navigation etc. and navigation in any direction if possible. The navigation mechanism is specific to various platforms. In android platform, all the navigation mechanisms are handled by navigation controller. The navigation drawer provides custom navigation interface. The navigation drawer is defied within the tag <android.support.v4.widget.DrawerLayout>, is shown below.

```
<android.support.v4.widget.DrawerLayout>
xmlns:android=http://schemas.android.com/apk/res/andro
id
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
        android:layout_height="match_parent">
</android.support.v4.widget.DrawerLayout>
```

Note: - For every activity\_XML file, there is a corresponding main activity java file or individual java activity file which contains all the required packages, methods, references and instances to perform the various actions and the corresponding listener methods irrespective to mobile native apps.

#### 4.8 Presentation Design

For providing interactive interfaces or environment presentation design is used in generic mobile native apps. The presentation design focuses on presenting a view in a native mobile app. It includes various transitions\_ animations to provide a view or a page in the mobile app. For presentation design transition and animations are used in various ways such as zoom in, zoom out, fade in, fade out, and card flip. For fade in operation. It is designed in XML and shown below.

```
<com.example.animationdemo.TouchHighlightImageButton[T
his button is used for displaying image instead of
text]
    android:id="@+id/thumb_button_1"
    android:layout_width="100dp"
        android:layout_height="75dp"
        android:layout_marginRight="1dp"
        android:src="@drawable/thumb1"
        android:scaleType="centerCrop"

    android:contentDescription="@string/description_image_
1" />[For zooming image_1]
```

Note: - For every activity\_XML file, there is a corresponding main activity java file or individual java activity file which contains all the required packages, methods, references and instances to perform the various actions and the corresponding listener methods irrespective to mobile native apps.

#### 4.9 Personification Design

The personification design allows the mobile native app user to share information with another user through social media tools such as Facebook, Twitter, LinkedIn and Google plus. It can be designed using different layouts, views and make it more interactive we can use different buttons and textviews. The following is shown below.

```
<ScrollView
xmlns:android=http://schemas.android.com/apk/res/androi
d
    android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="@color/grey_light">
```

Here “ScrollView” is used to encapsulate the contents that will be displayed in this particular page or view. The elements used here are linear layout, relative layout, buttons and text views. Scroll view is used here to adjust the list of users within the display, when a mobile native app user tries to connect with its friends or users which overfills the display size of the mobile device.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/buttonLayout"
    android:layout_below="@+id/card"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:gravity="center"
    android:background="@color/grey_light">
    <Button
    android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:text="Share"
        android:id="@+id/share"
        android:padding="8dp"
        android:background="@color/dark"
        android:layout_weight="1"
        android:textColor="#ffffff"/>
    </LinearLayout>
</ScrollView>
```

#### 4.10 Architecture Design

The architecture design concerns about different mechanisms such as mobility, sensors, input, output and memory mechanisms. This architecture design shown in fig.3, incorporates elements WLAN, GPRS, GPS, and browsers for mobility mechanisms in designing mobile native apps. It is also taking concern about different sensors such as location sensor, touch sensor, proximity sensor and temperature sensors etc. for sensing mechanism in designing the architecture of mobile native generic apps. The elements includes in the architecture design are only used for improving the versatility of the mobile native apps and these are configured into the hardware of the different platforms.

The elements for mobility mechanisms WLAN, GPRS, EDGE and Bluetooth. These elements are hardcoded into the hardware of the mobile device through establishing connections via adapters and sockets irrespective of any platform. The functionality of these mobility elements can be achieved by defining events and listeners for particular operation in mobile native generic apps. These are configured in the mobile device through protocols and their features are accessed by applications developed for this purpose irrespective of platform. There are various types of sensors such as position sensor, motion sensor, environment sensor and location based sensors (GPS). Mostly these sensors are built in mobile devices of any platform. The functionality of sensors are quite similar to the mobility mechanisms. Sensors are used in generic mobile native apps to perform specific operations like sensing temperature, getting location services, finding position of the mobile device and adjusting the orientation of the device.

The sensor framework is a part of hardware package irrespective of platform for any mobile native app. The sensor package consist of various classes and interfaces such as Sensor, SensorEvent, Sensormanager and SensorEventListener for utilizing sensor mechanisms. These interfaces include different methods for performing different actions and events in any platform, in mobile native apps.

Mobile devices incorporate memory mechanisms such as internal memory and external memory irrespective of platform. These memory mechanisms are configured by the mobile device manufacturer for running operating systems of various platforms and mobile native apps. The input and output mechanisms are provided by the system application of the mobile device irrespective of any platform for providing UI to the native app user. The functionality of the input and output mechanisms can be modified based on client requirements for developing mobile

native apps. The architecture design can also be included in other design phases of the proposed mobile native app design model.

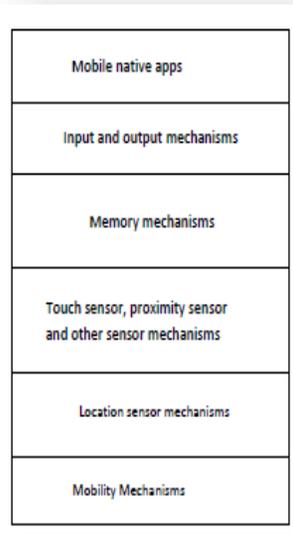


Figure 3. Architecture Design in Mobile Native Apps

### 5. Development Practices

Here the design phases such as user interface design, navigation design, presentation design and personification design can be developed and built irrespective of any platform. Here we have used android studio framework as the development environment supported by java language, text editor and the testing process is carried using an android emulator. We have already designed the phases using XML text editor as explained above.

#### 5.1 Developing User Interface

The user interface is built and developed through different layouts, views, widgets and wrapped through various containers. These design elements are developed through java programming languages in android studio frame work. Further, these design elements are implemented in genymotion emulator for validation purpose.

To develop widgets like edittext and buttons, which are designed previously in xml editor as shown in section 4.5, the widgets are defined in “EditText” and “Button” which is shown in figure 1. When we perform an action in button “change”, the onclicklistener() is activated and performs the necessary action for displaying the page or screen in mobile native apps.

The particular code segment comprises methods such as onCreate(), onView(),onClick() and corresponding listener methods onClickListner() and onViewlistner() for developing UI and many actions.

#### 5.2 Developing Navigation

To develop the navigation mechanism, a navigation drawer is

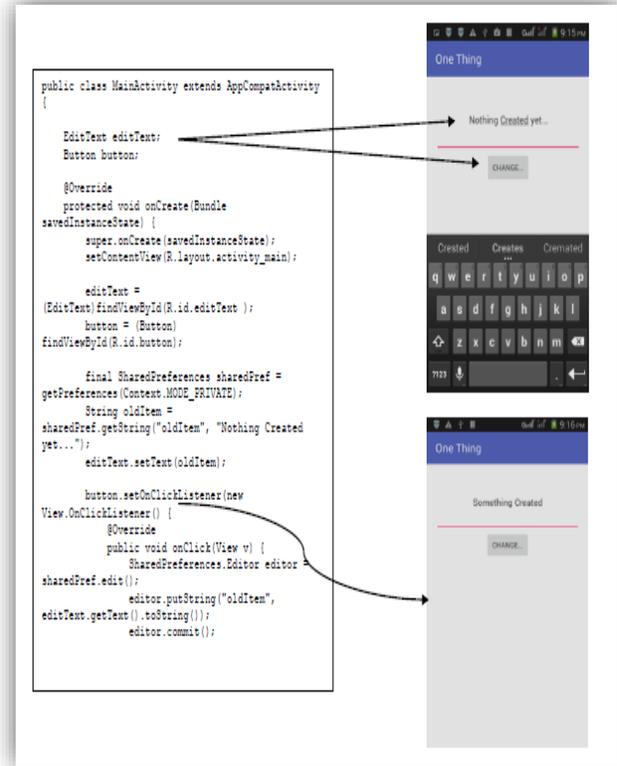


Figure 4. Developing User Interface in Mobile Native Apps

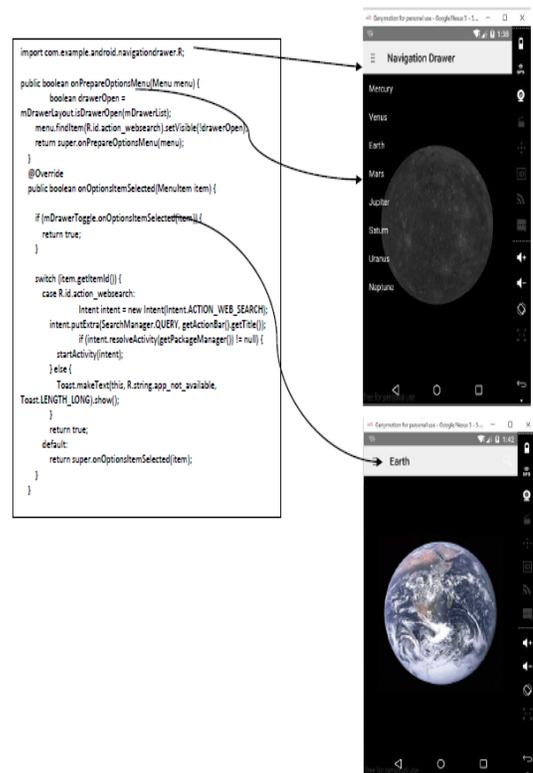


Figure 5. Developing Navigation in Mobile Native Apps

defined in the java activity file. The function of navigation drawer is to provide list of items present in the native mobile apps so that native mobile app user can select the desired option or item.

The navigation drawer holds the list of items through defining the method “onprepareoptionmenu()” where the items present in the mobile native apps. For selection of the desired item by the native app user, “onoptionsitemselected ()” method is used which is shown in fig.5.

### 5.3 Developing Presentation

For presenting a view in or a number of views in the generic native mobile apps, various transition animation operations are used. The transition animation includes zoom in, zoom out, fade in and fade out. The duration of the transition animation is defined by the mobile native app developers based on the client requirements.

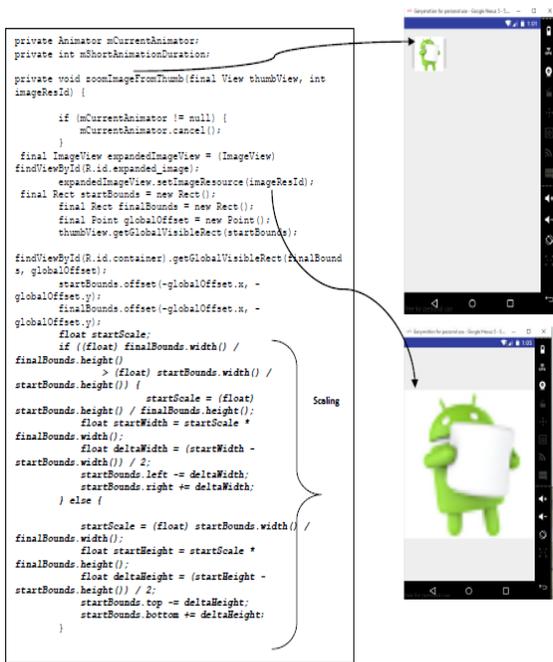


Figure 6. Developing Presentation in Mobile Native Apps

For transition operation “mCurrentAnimator” is used and its duration is set through “mShortAnimationDuration” which is shown in fig.6. For zoom-in operation the method ‘zoomImageFromThumb()’ is used as shown in the above code. For scaling the image into high resolution, “startScale” is defined and “xy” offset are set to expand the image where the image is zoomed according to the screen size of the mobile device for building the presentation design in mobile native apps.

### 5.4 Developing Personification

For achieving customization and personification in generic native mobile apps, the mobile native app user should be connected with other users through the social media tools so that mobile native app user can share and find its desired information about any domain. For sharing and finding information about any domain, the native app user must be connected with the social

apps. It can be by clicking on the “share” button as shown in fig.7, which prompts the user to the log in page of the mobile native apps. From the login page, which is shown in fig.8 the native app user logged into the desired social media tools and can find and share information about a generic domain.

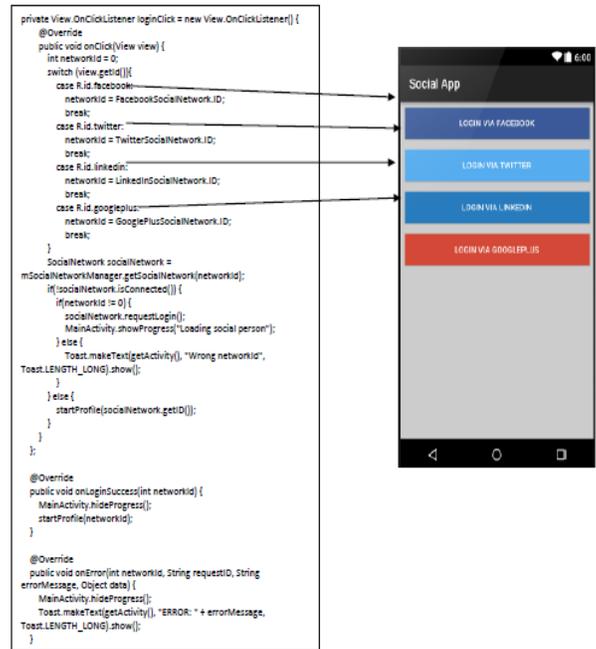


Figure 7. Developing Personification in Mobile Native Apps

## 6. Discussion

Here we have sent questions through Google forms to mobile native app developers of different software industries. After gating their responses, we have identified and proposed seven design phases for developing mobile native apps. The design model will work as a base for developing any mobile native app irrespective of any domain. At first, the user will analyze the content and generate the content for developing mobile native apps. For retrieving meaningful information, the users have to build ontology for a mobile native app. For building ontology, the user has to establish class axioms and property axioms among the classes for developing mobile native apps irrespective of any domain. The next phase is the user interface design, here the user will develop interfaces through using any platform for mobile native app customers. Here, we have done the user interface in android platform.

The next phase is the navigation design, here the user will build the navigation mechanism among the various activities through using any platform. Similarly the user will develop personification design by connecting social media tools through any platform. The architecture design provides the hardware components needed for building the mobile device and achieving mobility in it.

## 7. Conclusion and Future Work

In this paper we have proposed a design model for mobile native apps. This proposed model consists of seven phases as discussed above. We have tried to establish semantic relationship

between the attributes of mobile native apps in structural design model. In this conceptual design model, the attributes of mobile apps are designed semantically and structurally. We have find mobility mechanisms, input mechanisms and various sensors will be considered in the architecture design.

We have also identified the user interface design elements navigation design elements and presentation design element and social apps. These design elements can be designed in any framework like ios, windows and android etc and here we have taken android framework for design of these elements. In the personification design, any user can find and share its information about generic domain in mobile native generic apps. This proposed model will work as a generic standard for design and development of mobile native generic apps irrespective of any domain. Any user can design and develop the mobile native app in any domain following these phases of the proposed design model.

This model will solve various requirements of the users. If further the requirements of user changes, then changes will occur in all the design phases of mobile native apps and accordingly better design model will be built and developed. In the personification design, future work can be done on establishing chatting capability in the social media tools.

### **Conflict of Interest**

The authors declare no conflict of interest.

### **References**

- [1] A.Ginigi, S. Murugesan, "Web engineering: an introduction", MultiMedia, IEEE, **8**(1), 14-18, 2001.
- [2] D. Schwabe,G.Rossi," Developing hypermedia applications using OOHDm, In Workshop on Hypermedia Development Process, Methods and Models, Hypertext, **98**, 1998.
- [3] S.Ceri,P. Fraternali,A. Bongio," Web Modeling Language (WebML): a modeling language for designing Web sites", Computer Networks,**33**(1), 137-157,2000.
- [4] N.Koch,A.Knapp,G.Zhang," UML-based web engineering", In Web Engineering: Modelling and Implementing Web Applications ,157-191,2008,Springer London.
- [5] O.Pastor,J.Fons,V.Pelechano,"OOWS: A method to develop web applications from web-oriented conceptual models", In International Workshop on Web Oriented, Software Technology (IWWOST), 65-70, 2003.
- [6] A.Mushtaha,R.Tolba,"Integrating the V-Model into the Web Development Process".
- [7] <http://www.w3.org/TR/owl-guide/>
- [8] [http://www.google.co.in/?gws\\_rd=ssl#q=OWL+5.0+beta+framework+](http://www.google.co.in/?gws_rd=ssl#q=OWL+5.0+beta+framework+)
- [9] [http:// developer.android.com/guide/topics/ui/declaring-layout.html](http://developer.android.com/guide/topics/ui/declaring-layout.html)