

Force.com Applications

Building Data Model

1. Fields
2. Standard Fields
 1. Created when object are generated
 1. ID
 2. Name – Text OR Auto Number
 3. Owner
 4. Other – Created by, Modified By, Created date, last Modified date
 5. Data Types
 1. Numeric
 1. Number
 2. Currency
 3. Percent
 2. Calendar
 1. Date
 2. Date Time
 3. Limited Option
 1. Check Box
 2. Pick list
 3. Multi-select Pick List
 4. Formatted Text
 1. Email
 2. Phone
 3. URL
 5. Text
 1. Text
 2. Text Area (255)
 3. Long Text Area (32768)
 4. Rich Text Area (Formatted Text, Links, Images)
 5. Encrypted
 6. Calculation
 1. Auto Number
 2. Formula
 3. Roll Up Summary
 4. Custom Field Properties
 1. Name (__c)
 2. Label
 3. Universally Required
 4. Unique
 5. External ID – Custom Index added, Improves SOQL & Report Performance, Can have 3 External ID fields per object
 6. Default Value

- 7. Help Text
- 8. Field Dependencies
 - 1. Standard Pick list can be controlling not dependent
 - 2. Max values in controlling is 300 (so also if a field is both controlling and dependent)
 - 3. Before defining at least 1 value is required
 - 4. Custom multi select pick list cannot be controlling field
 - 5. Encrypted fields
 - 1. Provisioned feature
 - 2. View encrypted data permission
 - 3. Can edit even if not having View encrypted data permission
 - 4. Cannot be unique, external id or have default value
- 3. Object relationships
- 4. Lookup
 - 1. Child detail optional
 - 2. Each object can have as many as 25
 - 3. Creates a related list on the parent object
- 5. Master Detail
 - 1. Required child
 - 2. Delete parent – child deleted
 - 3. Master in one cannot be detail in another – not possible to create multiple level
 - 4. Can create between two custom or one custom and one standard (standard always master)
 - 5. Cannot create master detail to users or leads
 - 6. Detail side objects cannot have sharing rules, manual sharing or queues (no owner fields)
 - 7. Maximum 2 relationships per object
 - 8. Can perform cross object fields updates and roll up summary fields
 - 9. Can span multiple levels depending on Master object is standard or custom??

10. Special Relationships

- 1. Self relationships
 - 1. Lookup to self object
 - 2. Hierarchy
 - 3. May to many
 - 1. Two many to one relationships with same object on detail side
 - 2. Junction (Intersection object) object
 - 3. Deleted when one of the master is deleted (Kept in trash). If both master deleted it is deleted permanently.

11. Lookup Filter

- 1. Limit search results
- 2. Compare fields to other fields and standard value

3. Can be required or optional

Building User Interface

1. Custom Application
 1. Logo and Landing Tab
2. Custom tab
 1. Custom Object tab
 2. Web Tab
 3. VF Tab

Business Logic

1. Formula fields
2. Cross object formula
 1. Cannot reference cross object formula in roll up summary fields
 2. Cannot reference merge fields for objects related to activities
 3. Cannot reference record owner merge fields
 4. Limit for cross object formula fields is 10 unique relationships per object across all formulas and rules
3. Roll up summary fields
 1. Sum, Minimum, Maximum, Record Count
 2. Can create for all custom MD
 3. Some standard relationships also including Account Oppty and Oppty Product

Migrating configuration changes

1. Configuration stored as meta data
 1. Tools to move meta data
 1. Change sets
 2. Force.com ide
 3. Migration Tool
2. Change sets
 1. Point and click web interface to move metadata only between “**related**” orgs
 2. Requirements
 1. Apex code 75% code coverage unit testing
 2. Incompatibilities between versions – not migrated (can upload compatible part)
 3. Manually migrate changes that cannot be included
 4. **Renaming or deleting cannot be specified**
 5. Need change set profile permission
 6. Deployed entirely or not at all (if one component fails all fails)
 7. **Cannot modify once uploaded by can be cloned**

Analytics as a Service

Reports

1. Custom Reports Basics
 1. Reports do not save data only the report logic
 2. Data – Displayed based on access levels
 3. Reports are real time
 4. Reports access determined by folder
2. Report Builder
 1. Panes: Field, Filter and preview Panes
 2. Formats
 1. Tabular
 2. Summary – Sorting, sub Total
 3. Matrix – Grid, Compare related totals
 4. Scheduled reports run in the time zone of the user scheduling report
 5. To email a report it must be in public folder

Dashboards

1. Dashboard Component – based on report or chart
2. Based on data when dashboard was run
3. Dashboard components (20 max)
 1. Chart
 2. Table – 4 columns, summary fields
 3. Gauge
 4. Metric – Single no
 5. VF page
4. Chart Types
 1. Vertical column
 2. Horizontal Bar
 3. Line
 4. Donut
 5. Funnel
 6. Pie
 7. Scatter
5. Access
 1. Folder & Running user
 2. Each dashboard can have 3 filters
 1. Date/time, currency, picklist, lookup and text
 2. Fields that are common to all components
 3. Can not be added where you have VF components, used with bucket fields
 4. Dashboard Filter – based on 1 field (10 options)

Custom report Types (CRT)

1. Standard & Custom report Types
 1. Standard – Created when an object is created or relationship is created
 2. CRT
2. Creating CRT
 1. Define Primary Object
 2. Optional – Define report record set (related records / up to 4 objects) – with or without
 3. Optional – Add fields
 4. With (only when there is child) & With or Without (irrespective of child record)

Analytic Snapshots

1. Steps
 1. Create report
 2. Create object
 3. Create snapshot

Data Management

Data Management Overview

1. Uses Web Services API to access application
 2. Data Management operations
 1. Export
 2. Insert
 3. Update
 4. Upsert
 5. Delete
 6. Record Id
 1. Two Format 15 Digit Case Sensitive and 18 Digit Case Insensitive (first three char – object)
- i. Reports 15 Digits
 - ii. API 18 Digits
 - iii. For Updates APIs expect 18 Digit
 - iv. Can convert 15 to 18
 1. Ways to access record ID – URL, Reports, API, Formula Fields
 2. Help establish Object Relationships

3. Data Loaded in sequence depending on dependencies
4. “Inserting System Fields” like created date – can be set when importing if this feature is enabled (only during initial import of data). Cannot be updated.

1. Inserting System Fields is only for

i. Account

ii. Oppty

iii. Case

iv. Lead

v. Task

vi. Event

vii. All custom objects

1. Available only through API and backward compatible with SOAP based APIs

External Ids

1. UPSERT – An API function that provides INSERT & UPDATE in single call
2. External id
 1. Each object can have 3
 2. Help improve report and SOQL performance
 3. Upsert and External Ids are used for migration and data integration
 4. Can help auditioning the flow of information
 5. When UPDERTING If the extern ID matches with no record a new record is created, if it matches with 1 record – it is updated and if multiple records then error is thrown

Data Management Tools

1. Import Wizards
 1. 50000 records
 2. Accounts, Contacts, Leads, Solutions, Custom Objects
 3. Prevent Duplication (Using Account Name & Site OR Account ID, Contact Name, Contact Email OR Contact ID, Custom Object Name OR ID)
 4. API Based Tools
 1. Data Loader
 2. Can load more than 50000 records
 3. Schedule data loads – every night
 4. Export for backup

5. Delete multiple objects
 6. Data Loader
 1. Supports CSV files & External Database through JDBC
 2. Can run from command line
 3. SOQL – Salesforce Object Query Language
 4. Data Loader Operation
- i. Extract – uses SOQL
 - ii. Insert -
 - iii. Update – Matches records based on SF ID
 - iv. Upsert – Matches records based on SF ID OR External ID
 - v. Delete – Matches records based on SF ID
 1. Can be used only for DE, EE or UE
 2. Selecting Data Management Tool
 1. Dedup data – Import Wizard
 2. Schedule Data Load – Data Loader
 3. Load two related objects at once – Import Wizard
 4. Load < 50000 records – Import Wizard
 5. Export Data for backup – Data Loader
 6. Load data for object not supported by wizard – data Loader
 7. Upsert or Upsert with relationships – Data loader

Managing Data

1. Running Data Loader from Command Line
 1. java – jar DataLoader.jar
 2. com.salesforce.dataloader.process.PricesRunner with DataLoader.jar in classpath
 3. Options
- i. –Dsalesforc.config.dir=CONFIG_DIRECTORY.
 1. Data loader runs operation file or map in the configuration file
 2. If no directory is speceified takes the current directory
 3. Default directory
- i. C:\Program Files\Salesforce.com\Data Loader Version\config
 1. Batch process configuration file – process-conf.xml is at samplesconf directory
 2. If you use process-conf.xml setting process.name to the name of process specifies process to run otherwise config.properties is used for parameter settings
 3. Data Loader Encryption

1. Encryption Utility – RunBinencrypt.bat
 2. Command Line Prompts
- i. Generate a key
 - ii. Encrypt text
 - iii. Verify Encrypted Text
1. Mass Transfer Records
 1. To Transfer users need required user permission and read sharing access to record
 2. Can be used to transfer Accounts, Leads and Custom Objects

Designing Applications for Multiple Users

Design Considerations

1. Design Considerations
 1. Users – What they see and what they do
 2. Data Access path – Data that can be accessed by whom and restrictions
 3. Analyzing Business requirements
 1. Identify Actors and Data
 2. Identify How much each actor will see
 3. Specify access levels

Managing User Experience

1. Licenses
 1. User License
- i. Salesforce – Full access to standard CRM, Force.com Appexchange apps
- ii. Salesforce.com Platform – Only custom apps, Force.com Appexchange apps
 1. Feature License
- i. Marketing
- ii. Offline
 1. Profiles
 1. Define user's permission to perform different functions
 2. Profiles are associated with license type
 3. Work in conjunction with role hierarchy and sharing model

4. Profiles control permissions, access to data and user interface
 - i. Permissions – Admin & General, Standard Object & Custom Object
 - ii. Access to data – Field Level security
 - iii. User Interface – Page Layout, Apps, Tabs, Record Types
 1. Admin Permissions – View All data, Modify All data, Customize Application, API Only User (can not [login](#) to sfc.com, Password never expires)
 2. Standard (can not be created, deleted or customized) & Custom profiles (to modify the standard profiles – can be deleted or customized)
 3. Profile Permissions
 4. Permission Set
 1. Group of permissions in addition to their profile
 2. A user can have 1 profile and multiple permission sets
 3. 1 org can have up to 1000 permission sets
 4. Can only be used to give permission but not to deny
 5. Profiles v/s Permission sets
 1. Permissions sets – Applications, objects, tabs, fields, apex classes, service provider and VF pages
 2. Profiles – App Picker, Tab defaults, record Type, page Layout, Login hours, Desktop client access, category group access
 3. Field Level Security
 4. User Interface & profiles
 1. Record Type – View different details on page
 2. Page Layouts

Controlling Record Access

1. Considerations
 1. Ownership of records can be transferred to any user who has at least ‘read’ permission on object
 2. Child records in MD relationships inherit the properties of parent
 3. Users or Queue can be owners
 4. Levels of Record access
 1. Read only
 2. Read Write
- i. OWD
- ii. Role hierarchy
- iii. Sharing rules
- iv. Manual sharing

- v. Apex sharing
- vi. “view all data”
 - 1. Full Access
- i. Record Owner
- ii. Above in Role Hierarchy
- iii. “Modify all data”
 - 1. OWD
 - 1. Baseline level of access to data that we do not own
 - 2. Restrict access to data
 - 3. Three levels
- i. Public Read Write
- ii. Public Read Only
- iii. Private
 - 1. Roles & Hierarchy
 - 1. Record access through Role hierarchy
 - 2. Rolls up for all standard objects if the person above has access to object by profile
 - 3. Custom objects can be defined whether access can be granted or not (Use Grant Access using hierarchy in OWD)
 - 4. Public groups
 - 1. Default Public Group – All Internal users
 - 2. Made up of combination of
- i. Users
- ii. Roles
- iii. Roles & Sub
- iv. Other Public groups
 - 1. Use public groups to
 - i. Reduce no of sharing rules
 - ii. Folder access

1. Sharing rules
 2. Manual sharing
 3. Apex sharing Reason
- i. Developers define why a user has access to record
 - ii. Sharing reason does not create sharing rule
 - iii. Deleting a reason may delete sharing rule that uses that reason
 - iv. Requires 'Modify all data'
 - v. Used programmatically and not through app
 - vi. Sharing button will display reason
1. Access to Record Review is in following order
 1. Permissions (like View all data)
 2. Record Ownership
 3. OWD
 4. Role hierarchy
 5. Sharing (rules or manual sharing)

Designing Data Access Security

Implementing business processes

Automating Business Processes with workflow

1. Workflow rule – automate business process
 1. Process
- i. Select object
- ii. Specify criteria
 1. Only when Record is created
 2. Record is created or Updated and first meets criteria
 3. Every single time created or updates
- iii. Rule criteria
 1. Workflow actions
- i. Immediate

ii. Time dependent

1. Cannot be set for every time record is created or updated
2. Can monitor in Time dependent workflow q (can remove actions)
3. If a record has action pending in TDWQ and record no longer meets criteria or timing changes – the action will be updated.
4. If no longer meets – action is removed

iii. Action types

1. Email
2. Update field
3. Send outbound message
4. Assign a task

Auditing processes

1. Setup audit trail – track changes made to the setup
 1. Date, name of user and change
 2. 20 changes on screen and 180 days changes by download
 3. Field History Tracking
 1. History Related list
 2. Tracked for Accounts, Cases, Contacts, Contracts, Leads, opportunities, Solutions and Custom Objects
 3. Old value, new value, date time and user
 4. Long text area and multi-select pick lists are not tracked
 5. 20 fields per object
 6. Auditing tools
 1. Debug logs – Activities performed by end user or code
 2. Setup audit trail – config changes by admin
 3. Field history – data value changes

Approval processes

1. Steps
 1. Process definition – which records enter process, settings for whole process
 2. Initial submission action – locking, assigning task, email, update field, ob message
 3. Step definition – which records enter step (all or some), multiple step

i. Decision criteria

ii. Assignment

1. Final rejection action – unlock, task, email, update fields, ob message

2. Final approval action – unlock, task, email, update fields, ob message
3. Recall action – unlock, task, email, update fields, ob message
4. Process visualise
 1. Read only visual
 2. Workflow v/s Approval
 1. Wf triggerd when record is saved – approval manual
 2. One set of criteria and actions – have multiple steps, entry and step criteria and step actions + more actions
 3. Can be modified or deleted – some attributes can not be modified without deactivation.
 4. Skipping steps
 1. Criteria to Skip Step in an approval process
 2. Can be a filter criteria or formula
 3. Options

i. Go to next step (only if there is next step and if next step is deleted SF will automatically change to reject record)

ii. Approve

iii. Reject

1. Selecting Go to next step in the first step and record does not meets criteria – record is rejected
2. Parallel approval process
 1. Can send approval to multiple users
 2. Up to 25 users
 3. Unanimous / based on 1st response
 4. Outbound messages
 1. Designate end point
 2. Secure configurable API message in xml format
 3. Dynamic approval process
 1. Route records based on approval matrices
 2. Approval request to users listed in record
 3. Creating

i. Create lookup fields on object approved

ii. Custom object as approving matrix

iii. Populate approval matrix

iv. Create apex code to fill lookup fields from approval matrix

v. Approval process to utilize lookup fields

1. Order of automated process
 1. Validation rule
 2. Assignment rule
 3. Auto response rule
 4. Workflow rule
 5. Escalation rule
 6. Visual workflow (flows)
 1. Required Manage Force.com Flow
 2. Cloud Flow Designer

Preserving Data Quality

1. Validation Rules – data meets the standards (formula or expressions)
 1. Run on server
 2. UI, API, Import wizard
 3. You write – Error condition formula and Error Message
 4. Conditionally required fields
 5. Data Quality
 1. REGEX – compares text field to Regular expression (pattern used to describe a set of strings according to syntax rules)
 2. Data consistency
 1. VLOOKUP – validation rule stored in another object
 2. Prevent data loss
 1. Use in conjunction with Rollup summary field
 2. Parent – sums no of child objects

Visualforce Pages

Visualforce Introduction

1. Two types of UIs
 1. Page Builder
 2. Visualforce
- i. Full control over UI
 1. In line editor – available on “Development mode”
 1. Automated completion, syntax highlighting, documentation
 2. Create a page – /apex/pagename
 3. Visualforce Pages – defined using mark-up language
 1. Can include
- i. Visualforce tags

- ii. Force.com expressions
 - iii. HTML / CSS
 - iv. Javascript
 - v. Flash
 - 1. Can Ajax Toolkit to access web services API to connect to SF data
 - 2. Limited to 15 MB
 - 3. Visualforce component tags – generated HTML tags behind the scene to create the page
 - 4. Benefits
 - 1. Understand SF metadata
 - 2. Coupled with Force.com – same performace
 - 3. Auto upgrade
 - 4. Separate view from navigation control and data model
 - 5. Confirms to MVC
 - 6. MVC
 - 1. Model (data Structure)
 - 2. View (data display)
 - 3. Controller (rules)
 - 4. Elements of VF
 - 1. Pages
 - 2. Components – referenced components for reuse
- i. Standard (prebuilt constructs) or Custom
- ii. Referenced using a Tag Library Model
- iii. Dynamic Visualforce Components (Designed in Apex – Render pages based on variety of states)
 - 1. Controllers
 - i. **Standard** (new, edit, save)
 - 1. <apex:page standardContoller=”Contact”>
 - ii. **Custom**
 - 1. Custom controller – implemented in Apex
 - 2. <apex:page Contoller=’XX’>
 - iii. **Controller Extensions** – OOB functionality while creating custom actions and data references

1. <apex:page Controller='XX' extensions=MyClass,MyOtherClass">
- iv. Logic and data references
- v. A VF page reference one main controller
- vi. VF uses Expressions syntax to bind data to controllers
 1. All data in {!...} is evaluated as expression
 2. Versioning
 1. Previous versions are available

10. Namespaces

1. Standard Tag – <apex:page>
2. Custom Tag – <c:customtag>
3. Register Custom namespace – <thirdpartynamespace:customtag>

11. Ways to add VF pages to Salesforce UI

1. Link to URL
2. Override standard buttons
3. Overriding Tab overview pages
4. Custom Tab
5. Embedding on Page Layout
6. Adding to dashboard
7. Custom help to custom object

Visualforce tags

1. Tags
 1. Tags
 2. Text directly on page
 3. HTML tags and Comments <!--Comment -->
 4. Can use Javascript but not Javascript comments
 5. Hierarchical tags – well formed – means start and end tag or self contained tag
 6. Visualforce component tags – similar to JSP tags
 7. Tag Binding
 1. Three types of Tag bindings
- i. Data – Use Expression Syntax to pull data made available by page controller
 1. API naming syntax
 1. __C
 2. Custom objects referenced through relationships – dot notation and __r
 3. Expression Syntax

1. Dynamic Object Data {!..}
2. Global Data – {{\$.}}
3. Local variables can be created

ii. Action – Use Expression Syntax to call action methods for functions coded in page controller

iii. Component – does not use expression syntax and uses component attribute values to reference other component Ids

Basic Page Components

1. Attributes and values
 1. Values Types

i. Integer

ii. String

iii. Object {!Account}

iv. Action Method – {!Save}, {!Edit}, {!Delete}

v. Boolean – true, false

vi. Iteration

1. Layout Components

i. <apex:page> – single VF page

1. standardController or controller
2. title (String)
3. tabStyle (String)
4. action (Action Method)
5. renderAs (string)

ii. apex:variable

1. value (Object)
2. var (string)
3. Static Resources – Type of salesforce storage, like archive, images, CSS

i. Referenced in code using \$Variable

ii. Limit 5MB file overall 250 MB

iii. Faster upload – caching

iv. Usage

1. <apex:flash>
2. <apex:image>
3. <apex:stylesheet>
4. <apex:stylesheet>

i. Stylesheets located in /sCSS/ directory with dStandard.css and allCustom.css

1. Pageblock tags

i. <apex:pageBlock>

1. Mode (string) – detail / edit
2. Title(string)
3. tabStyle(string)
4. facet tags

ii. <apex:pageBlockButtons>

1. location

iii. <apex:pageBlockSection> – section with columns

1. Collapsible (Boolean)
2. Columns (integer)

iv. <apex:pageBlockSectionItem>

v. <apex:SectionHeader>

1. Title
2. Subtitle
3. Help
4. Other UI Tags

i. <apex:toolbar>

ii. <apex:toolbarGroup>

iii. <apex:tabPanel>

iv. <apex:tab>

- v. <apex:panelBar>
- vi. <apex:panelGrid>
 - 1. Template Tags
- i. <apex:composition template:MyTemplate>
 - 1. Coarse Metadata Components
 - 1. Apex:detail
- i. Subject
- ii. relatedList
- iii. relatedListHover
 - 1. apex:relatedList
- i. List
- ii. Subject
- iii. pageSize
 - 1. apex:ListView
- i. type
 - 1. apex:enhancedList
- i. customizable
- ii. type
- iii. rowsPerPage
- iv. width
 - 1. apex:repeat
- i. value
- ii. rows (integer) (0 all rows)
 - 1. Chater

- i. showChatter on apex:detail
- ii. <chatter:feed>
- iii. Chatter:follower
- iv. Chatter:follow
- v. Chatter:feedwith follow

- 1. Message tags

- i. Apex:message
- ii. Apex:pageMessage

Visualforce Form & Output Components

- 1. Form – enter data

- 1. **a. Apex:form**

- i. Accept
- ii. Title
- iii. Target

- 1. **b. Apex:inputfield**

- 2. **c. Apex:inputwidget**

- i. Widgets that do not correspond to data in SF (have limitations)

- 1. Inputcheckbox
 - 2. inputHidden
 - 3. inputsecret
 - 4. inputtext
 - 5. inputtextarea

- ii. use when you want to vary size of data, tab sequence, required inputs, bind to custom controllers and standard javascript events

- 1. selectWidgets (display of UI widgets in organized tables)

- i. selectcheckboxes
 - ii. selectList

- iii. selectRadio

- 1. apex:inputfile (attachments, documents, blobs)

- i. accept

- ii. contentType

- iii. filename

- 1. apex:commandButton & apex:commandLink (used within form tags)

- i. action

- ii. value

- 1. Output (display without allowing change in data and most have parallel form components)

- 1. Apex:Outputlabel

- i. Vales

- ii. For

- iii. Tabindex

- 1. Apex:outputField

- i. Value

- 1. Apex:outputField

- i. Value

- ii. Title

- iii. coords

- 1. Apex:outputLink
 - 2. Apex:OutputPanel

- i. Set of content grouped together (eg for partial page refreshes using ajax)

- 1. Apex:outputText
 - 2. **g. Apex:pageBlockTable**
 - 3. **h. Apex:dataTable**

4. **i. apex:dataList**
5. apex:column/
6. **k. apex:flash**

- i. src
- ii. height
- iii. width
- iv. flashVars
 1. apex:facet

- i. to override headers, footers and captions

Visualforce components for modularity

1. Custom Components
 1. Can be standalone or accompanied by custom controller
 2. **b. Apex:component**
 3. Access using **<c:component>**
- i. Controller or extensions
- ii. Access (global (outside namespace) or public)
 1. Apex:attribute
 2. Apex:componentBody
 3. Page inclusion components
 1. <iframe>
 2. <include> – include VF page in iframe
 3. Template Tags
 1. Reusable components for pages – apex reference page and target VF page
 2. Three tags
- i. **<apex:define>** – include dynamic content to be inserted
- ii. **<apex:insert>** – insert named area defined by define tag
- iii. **<apex:composition>** – container that references name on the target page
 1. Performance considerations
 1. One <apex:form> on one page
 2. Cache resources
 3. Page size < 15MB

4. Increase time interval to call apex from VF page to reduce delays caused by concurrent requests
5. Remove unnecessary fields esp the one that have MD relationship with other objects or are large text

Javascript in VF pages

1. Action Binding & java Scripts
 1. Only actions that are shared across all objects are exposed through standard controllers
 2. For others – Standard SF controllers are using javascript and expression syntax with **!URLFOR** and **\$Action**
 3. Javascript **functions**
 1. Can be included in header `<script>` or in a .js file
 2. **b. Can be called any time after declatred**
 3. Efficient way
 4. Ajax
 1. Used for partial page referesh
 2. Has been used to access Web services API within Javascript
 3. Due to data binding capabilities of VF the need of ajax for API is not required
 4. Ajax Tags
 1. **actionStatus** – start stop status of ajax request
 2. **actionSupport** – call second component when event happens to first component
 3. **actionPoller** – similar to actionSupport but event is based on timer
 4. **actionFunction** – support for invoking controller action from javascript code
 5. **actionRegion** – demarcate the part of page to process
 6. partial Page update
 1. **reRender** attribute

i. use along with `<apex:outputPanel>`

1. Ajax

i. Status update using `actionStatus`

ii. `StartText` and `StopText` to display text ate the start and end of update

iii. Display text or graphics after referesh `<apex:facet name=”stop”>`

1. Ajax & Apex web service

1. Create an apex class and expose it as a web service
2. Call web service from a VF page

Advanced VF topics

1. Force.com Sites
 1. Unauthenticated sites
 2. Use cases

- i. Web applications – Event registration / reviews

- ii. Business applications – Recruiting Portal

- iii. Extend CRM – Web to lead forms / Landing Pages

- iv. Corporate site – website / intranet

1. Mobile devices
 1. Salesforce mobile – BB, iPhone, Windows
 2. iPhone better Javascript support than BB
 3. Create pages for mobile devices – separation & redirection, lowest common denominator, conditional code
 4. Messaging:emailTemplate
 1. emailHeader
 2. htmlEmailBody
 3. plainTextEmailBody
 4. attachment
 5. Chatter Data Model
 1. Core object – FeedItem
 2. Chatter Tags
 1. showChatter
 2. <chatter:feed>