Survey On Application Software

- Industrial Insights -

Report By

Aarthi Babu

University Of Northern British Columbia
Computer Science

Contents

Chapter 1

Introduction

Application Software is program or group of program customised to perform group of activities for end user. Generally software are classified as system software and application software.System software consists of low-level programs that is designed to run computer's hardware and application programs like compilers, loaders, linkers and so on. Application software resides above system software like database programs, word processors, spreadsheets. In this survey, we focus on the architecture and process involved in development of different types of application software.

Software application architecture is the process of designing a structured solution that focuses on how the components in the applications interact with each other, while optimizing common quality attributes such as performance, security, and manageability.The architecture are structured with consideration of user and the business goals.The selection of data structures and algorithms are design concerns which would overlap with architecture concerns.These scenarios where the both of them overlap are discussed in detailed in later chapters.The architecture must be flexible to handle the changes in software or hardware technologies and requirements that are not known in the early stages of design process as the design of the application will evolve during the development stages.

1.1 Command-Line Application

Technology has come long way since the first computers were created, back around the start of World War II. The first generation of software application are command line programs which are mostly single command at a time and uses it to accomplish all the application requirements in that particular loop. These programs are shared as binaries(executables) which can be compiled from the source code, specific to the architecture.

1.2 Desktop Application

In spite of early application which was designed to run from mainframe computers and accessed through terminals devices, the proposal of powerful desktop applications which can be run in the local personal computer dethroned the previous generation application software. The mainframe was replaced with server by the client server model which induced distributed application structure that partitions tasks or workload between the service provider (Server) and the service requester (Client).

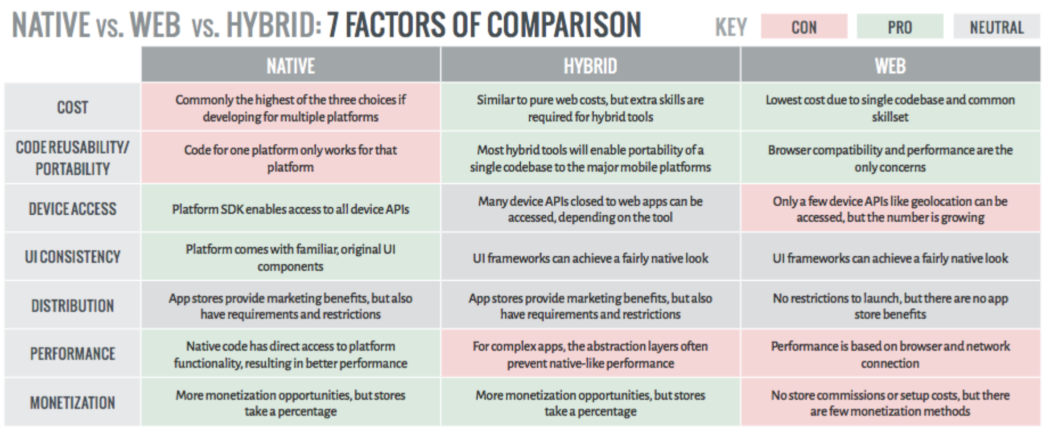
1.3 Web Application

Steady enhancements in hardware specifications and broadband speeds led to major improvements in the quality and quantity of WWW content. Websites inflated their features beyond static web pages by becoming more interactive with the increase of multimedia content. As browsers and development platforms evolved, and more and more people began to use the internet and email, more businesses established their presence in the online world. These businesses leveraged the emerging interactive capabilities of the web to introduce applications that were served directly to a web browser, and these web applications became very popular. Any Web application is a client server application which consist of a client inform of browser and a web server. The business logic of the application is distributed among the server and client where information is exchanged through a channel and the data is stored in the server end.

1.4 Mobile Application

Nowadays, an ever-growing percentage of portable device like smart phones have doubled the number of users accessing the internet on their smart phones.The need of mobile applications became pretty obvious. Mobile applications should be designed in such a way that the power consumption and processing power is reduced. Android and iOS being the major players in the industry, they natively support java and ObjectiveC as the native language.

The mobile applications can be categorised into mobile web application, native application and hybrid application.A native application is developed for a certain mobile device (smart phone, tablet, so on). They're installed directly onto the device. Users typically acquire these applications through an online store or marketplace such as The App Store or Android Apps on Google Play.It was started working as standalone, and in the recent years, we see a lot of integration with the web which is named as hybrid application.


NATIVE VS. WEB VS. HYBRID: 7 FACTORS OF COMPARISON

KEY: CON | PRO | NEUTRAL

	NATIVE	HYBRID	WEB
COST	Commonly the highest of the three choices if developing for multiple platforms	Similar to pure web costs, but extra skills are required for hybrid tools	Lowest cost due to single codebase and common skillset
CODE REUSABILITY/ PORTABILITY	Code for one platform only works for that platform	Most hybrid tools will enable portability of a single codebase to the major mobile platforms	Browser compatibility and performance are the only concerns
DEVICE ACCESS	Platform SDK enables access to all device APIs	Many device APIs closed to web apps can be accessed, depending on the tool	Only a few device APIs like geolocation can be accessed, but the number is growing
UI CONSISTENCY	Platform comes with familiar, original UI components	UI frameworks can achieve a fairly native look	UI frameworks can achieve a fairly native look
DISTRIBUTION	App stores provide marketing benefits, but also have requirements and restrictions	App stores provide marketing benefits, but also have requirements and restrictions	No restrictions to launch, but there are no app store benefits
PERFORMANCE	Native code has direct access to platform functionality, resulting in better performance	For complex apps, the abstraction layers often prevent native-like performance	Performance is based on browser and network connection
MONETIZATION	More monetization opportunities, but stores take a percentage	More monetization opportunities, but stores take a percentage	No store commissions or setup costs, but there are few monetization methods

Figure 1.1: Native vs Web vs Hybrid

Internet-enabled applications that have specific functionality for mobile devices are called as mobile web application. They are accessed through the mobile device's web browser (i.e. on the iPhone, this is Safari by default) and they don't need to be downloaded and installed on the device.Although mobile websites and mobile apps aren't the same thing, they generally offer the same features that can help grow business by making it easier for customers to find and reach it.

Chapter 2

Software Application Architecture

Overall structure of the software application can be represented in the logical grouping of components into layers that interact with each other according to the functionality and features of the application. This chapter will focus on how the application is divided into components and the services provided by each layer.These layers help to uniquely identify the different kind of task performed by each components. Each layer would consist of multiple sub layers which performs specific type of task which would greatly help while debugging.

By analysing the types of components that exist in most solutions, you can construct a meaningful map of an application or service, and then use this map as a blueprint for your design. Dividing an application into separate layers that have distinct roles and functionalities helps you to maximize maintainability of the code, optimize the way that the application works when deployed in different ways.Figure 2.1 represents the highest and most abstract level of the software architecture.It gives clear perspective of theirs interaction with users, relationship with other application that requests the services implemented within the application business layer,web services hosted by the application , data sources and the remote service used by the applications offered by other software.

2.1 Basic Architecture Layer

The common three layer design consist of Presentation Layer ,Business Layer and Data Layer.

- **Presentation Layer :** This layer consist of two sub components which contains purely user based functionality that manages the user interaction with the application. This layer acts as a bridge between the user and business logic layer. User Interface and user interface logic are the two major components found in this layer.This layer is responsible for the user input and display

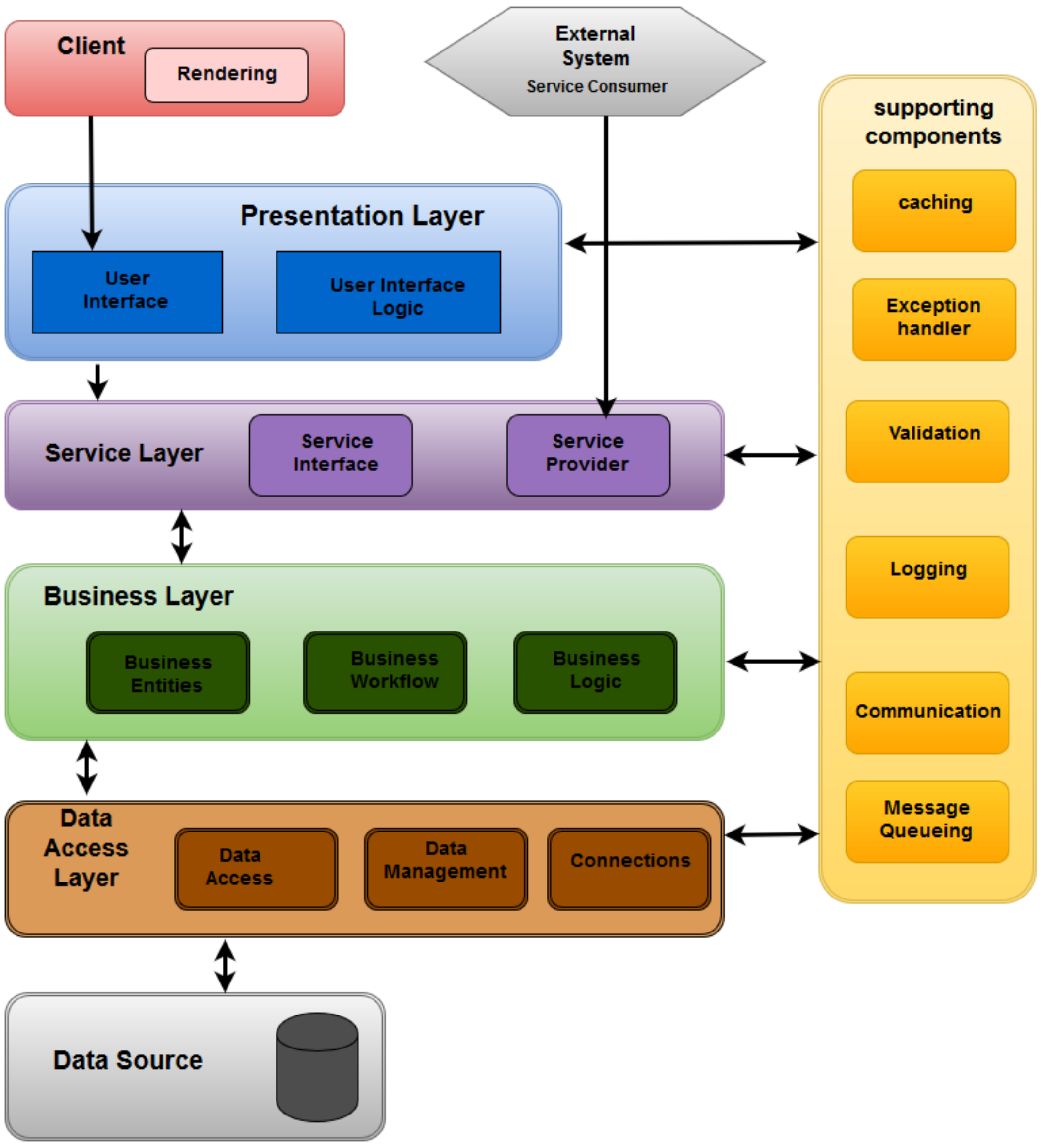


Figure 2.1: Software Application Architecture

in addition to the components that organise user Interaction.To make the code modular the implementation of visual elements which would display data to user and accepts input is separated from the presentation logic which gives rise to the idea of model, view and controller.To handle the design based issues , security issues and improve the application performance and user interface responsiveness, the extra components are introduced which discussed in forth coming chapters.

- **Business Layer :** This layer implements core functionality of the application and is concerned with the processing, transformation, retrieval and management of data.The core implementation include the business rules,entities and work flow of the rules. Application façade is a optional component which acts a simplified wrapper on business logic components by combining multiple operations into a single operation in a logical way. This reduces dependencies and improves modularity. This prevents the external service request from knowing the details of the business components and the inter relationship between the operations.

- **Data Layer :** Data access logic and data store are the major components of this layer. Much more components like object document mapper , object relational Mapper etc which deals with translation of data can be introduced into this layer based on the requirements.This layer provides access to data within the boundaries of the system and the data exposed by the other networked systems through web service.Data access logic components forms a interface that the components in the business layer consumes.

2.2 Requirement-Based Components

Some layers are pluggable in the architecture on requirement basis.When an application must provide service to external system , the service layer is used to expose the business functionality without exposing the operation signature.There are components which are added to the architecture to handle issues.The issues and the corresponding layer which handles the issue can be categorised based on the architecture layers.

2.2.1 Issues and Add-on - Presentation Layer

- **Caching :** Caching is considered to be a best mechanism to improve application performance and user interface responsiveness. Caching is used in presentation layer to avoid frequent data lookups which in turn avoids network round trips and to store the result of the repetitive process to avoid the unnecessary duplicated processing. Its important to make the cache thread-safe when it is used in multi-thread environment.

- **Exception Management :** Employing a centralised exception management is considered to be a consistent way to manage the unexpected exceptions. The exception which propagates across layers are blocking issue which would crash the application. Differentiating the error occurred into system-based or business logic based will be helpful to provide a friendly error message to the users. If it is business errors,a error message can be displayed and allow user to re-try that particular operation. In case of system error, a error message can be displayed along with the troubleshooting assistance. It is important to ensure no sensitive data is exposed in the error pages, error message and log files.

- **Compositions :** User interface composition patterns and templates supports the creation of the views and the presentation layout at the run times.It is easy to develop and maintain if the presentation layers uses independent modules and views that are composed at run time. These templates helps to minimise the code and library dependencies that would otherwise force recompilation and redeployment of the modules when the dependencies are upgraded.

2.2.2 Issues and Add-on - Business Layer

The business layer includes the previous add-ons like caching

Data Layer Batching : To achieve high performance , the similar queries are batched avoiding multiple database request and execution. Each database request is considered to be a overhead. Batched commands reduces the round trip time to database and minimizes network traffic as well as the size of the connection pool.

- **Authentication :** Security and reliability of an application is important for business layer where authentication comes into picture.If your business layer will be used by multiple application , plugging an authentication mechanism into the business layer ensures the security of the sensitive data. In case of client server application, the application layer lies in client and the business layer lies in server application. Both application communicates over network since they are deployed in different machines. In this case, a centralised platform which handles authentication is mandatory for the reliability of the application

- **Authorization :** In most of the applications the authorization components manages issues like information disclosure and user privileges for specific activity.Authorization are used to allow specific group of user to perform some sensitive business information and view some sensitive data. It promotes hierarchy of user privileges and the related operations ensuring security of the data sources.It is important to implement the authorization code in a modular way which doesn't impact on performance of the application and doesn't have any dependency with business layer .

- **Logging and Auditing :** Maintaining a centralised logging and auditing system would be useful in tracking the issues in production and the illegal user actions in important situations. System Monitoring tools are used to identify state and performance of the application. Logging helps to track this performance and the load handled by the application. It is important to ensure that logging failure doesn't affect the functionalities of business layer.

2.2.3 Issues and Add-on - Data Layer

- **Batching :** To achieve high performance , the similar queries are batched avoiding multiple database request and execution. Each database request is considered to be a overhead. Batched commands reduces the round trip time to database and minimizes network traffic as well as the size of the connection pool.
- **Connection :** Establishing connection to data sources is considered to be basic part of data access layer.Creating and maintaining data sources connection in inefficient way directly impacts the performance of the applications. Connection pooling within a security model would serve the purpose in efficient way.Holding connection for short period and avoiding the usage of user DSN or system DSN would help to achieve high performance and security.
- **Object Relational Mapping (ORM) :** When the application is designed using object oriented model, It is difficult to translate the relational model to business entities. ORM can be used to create schema which supports the object model and provides a mapping between the database and domain entities.When working with Web applications or services, group entities and support options that will partially load domain entities with only the required data—a process usually referred to as lazy loading. This allows applications to handle the higher user load required to support stateless operations, and limit the use of resources by avoiding holding initialized domain models for each user in memory.

2.2.4 Issues and Add-on - Service Layer

In case of external application acting as service consumer or the application is designed in client server model, the service layer acts as interface between the systems. Service Interface represent the operations supported and their associated parameters and data transfer objects. To avoid multiple call over network, the service interface are designed to serve multiple operations grouped based on business logics. Authentication is used to determine the identity of the service consumer.Authorization is used to determine which resources or actions can be accessed/possessed by an authenticated service consumer. Communication component can be designed based on the requirement like request-response ,duplex communication and asynchronous communication.Message queueing can be used to handle tremendous burst of asynchronous message .

Chapter 3

Presentation Layer

The design of presentation layer begins by identifying the potential customer base, and understanding the objectives of the customer and task the user wish to accomplish when using the application. As the main focus of the application design must be user centred, the sequencing of tasks or operations should be designed as per the user expectation. It may be a structured step-by-step user experience or unstructured experience where they can perform more than one tasks simultaneously. One important aspect that has great influence on the choice of technology is functionality required for the user interface. Prioritizing the requirements like rich functionality, user interaction, user responsiveness , user interface , personalization requirements and graphical support will help to choose User Interface Type.

Most of the user interface requirements needs more than one UI type.The frameworks, tool and implementation language used for UI requirements differ according to the application platform.

3.1 Mobile Application

Mobile application can be classified as thin client and rich client application.Rich client is designed to support the native or hybrid applications which supports offline or intermittent online scenarios. Web or thin clients supports only online scenarios.Resources constraint should e taken into account while designing user interface for mobile application.

Many developers or vendors want to write an application once using single codebase , the run that application in multiple platform with a very small change in codebase. This approach is famously known as Write Once, Run Anywhere (WORA).WORA removes the redundancy in development at the cost of user experience and application performance.This application is called as hybrid mobile application which has native application experience. Improving the performance and user experience are in future roadmap.

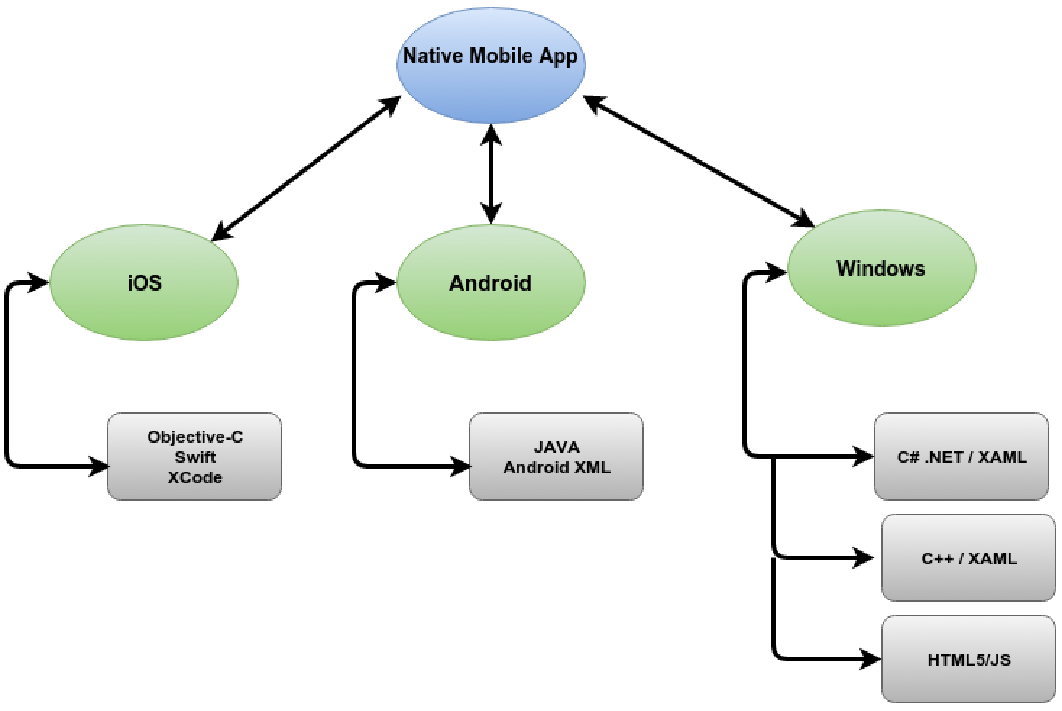


Figure 3.1: Languages - Native Moblie Application

3.1.1 Apache Cordova - Frameworks

Apache Cordova is an open-source mobile cross-platform development framework which allows you to use standard web technologies - HTML5, CSS3, and JavaScript . In mobile development, this framework extends more than one platform avoiding the implementation time taken for each platform.Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's capabilities such as sensors, data, network status, etc. It is flexible in developing application which has combination of native application components with a web view that can access device level APIs using Core Plugins.

3.1.2 Ionic - Frameworks

Ionic is a free open source framework facilitates the development of hybrid native mobile using web technologies like HTML,Javascript and CSS. Its core is built up with AngularJS javascript framework and supports SASS (Syntactically Awesome Style Sheets) CSS extension . This features qualifies Ionic as a unique framework.It simplifies development and testing of the application by providing client side model view controller architecture. User Interface response is pretty faster than Backbone and Knockout and it possesses a large number of 3rd party plug ins and extensions. Currently Ionic supports iOS and Android devices, Windows phones and FirefoxOS support are considered to be in future roadmap. This framework can be only used for hybrid mobile application.Ionic is built on top of Apache Cordova which is open source mobile development framework.

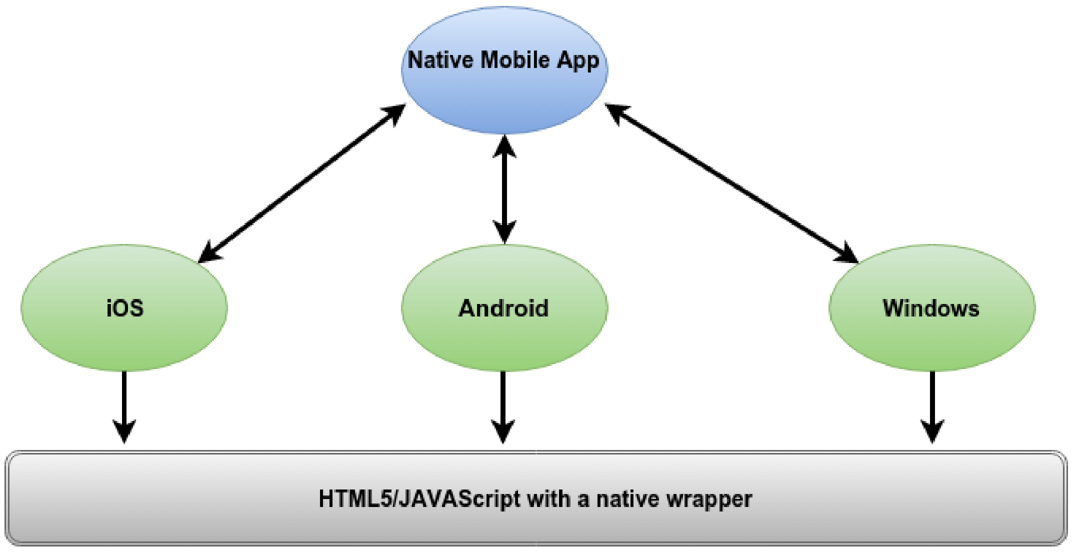


Figure 3.2: Cordova Languages - Hybrid Moblie Application

3.1.3 Mobile Angular UI - Frameworks

Mobile Angular UI is an HTML 5 framework which uses bootstrap 3 and AngularJS with better user response. In other words it combines the best of the web framework which enables you to create hybrid application and web application. This framework works well with all browsers including older versions and offers feature of customizing build workflow with GRUNT. Grunt is used run automation using a task runner with zero effort.

3.1.4 Xamarin - Frameworks

Xamarin applications are written in C# and compiled to binaries compatible with native device giving access to all the device API through C# wrappers.To develop application that look like native, Xamarin uses native UI mechanisms for each platform. Recently Xamarin.Forms was released which enables the use of common UI technology, XAML and work on cross platforms using native controls. Xamarin takes advantage of all the third party libraries written for .NET stack

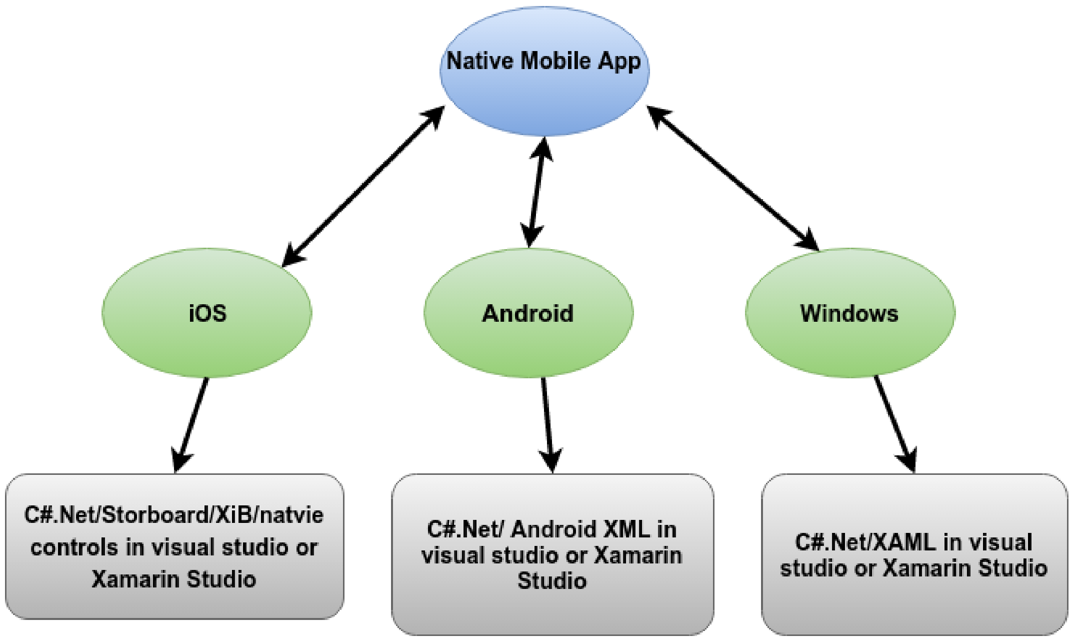

Native Mobile App
iOS
Android
Windows
C#.Net/Storboard/XiB/natvie controls in visual studio or Xamarin Studio
C#.Net/ Android XML in visual studio or Xamarin Studio
C#.Net/XAML in visual studio or Xamarin Studio

Figure 3.3: Xamarin Languages - Hybrid Moblie Application

3.2 Web Application

Server-side HTML, JS generation widgets (AJAX) and Service-oriented single-page are the three Types of web application . Server-side Html was used as information portal with less user interaction as the server generates HTML-content and sends it to the client as a full fledged HTML-page. JS generation widgets (AJAX) was introduced to promote read-write web which contains user generated content and improved the more user interaction. These web applications used Ajax programming which uses javascript to upload and download new data from the web server without full page reload. Some applications where programmed in Flex which helped to populate huge amount of data and other heavy user interactions.The application programmed in Flex are compiled and displayed as Flash within the browser. The foremost advantage is that updates from the server arrive only for the part of the page requested by the client. A particular widget is in charge of a part of the page; changes in a part will not affect the whole page.

Later Web application architecture undergone some changes where part of functionality is shifted to the client-side. An HTML-page which contains JavaScript-code is downloaded for addressing particular web service. It retrieves only data from the back end which is used by JavaScript to update the html content of the page.In this type of application , the responsiveness is the high as UI is generated in client side with necessary variants.Under this architecture this criterion has the lowest influence from the server side. The server only has to give the JavaScript application to the browser. On the client side performance and browser type are of the biggest importance.

3.2.1 Bootstrap - Framework

Bootstrap is a open source front-end framework developed by twitter. It consist of few ready to use CSS and HTML based design templates which aims to ease the development of dynamic websites and web application. The responsive grid system of the framework empowers the efficient layout of the application. Bootstrap is uniform accross all the platforms and it supports browsers like Chrome (Mac, Windows, iOS, and Android) Safari (Mac and iOS only) Firefox (Mac and Windows) Opera (Mac and Windows) IE8+.

3.2.2 Foundation - Framework

Foundation is one of the open source front end framework which contains many tools useful for making responsive dynamic websites and web application. Basically it is built with HTML, CSS, jquery and can be extended to use in conjunction with Sass and Rails.Features and components may look similar to bootstrap, but there are some slight difference like off-canvas navigation , right-to-left support etc.Foundation provides better environment for customization than bootstrap.

3.2.3 AngularJS - Framework

AngularJS is a open source framework used to develop dynamic web applications. Angulars includes two way data binding and dependency injection. Major percent of the web application code base is dedicated to traversing, manipulating, and listing to the DOM. Two Data-binding handles the synchronization between DOM and the model automatically makes required implementation to disappear . AngularJS incorporates the basic principles of MVC software design pattern which refers to Model-View-viewModel(MVVM) pattern.

TO DO // More Info

3.3 General Tools

3.3.1 ReactiveX - libraries

Tremendous increase in need to respond and trigger actions based on data events in mobile application expanded the requirement of handling events in asynchronous manner. Component Object model was one of the previous technology that was used to execute asynchronous tasks. Recently Netflix introduced ReactiveX library which to process asynchronous task with robust architecture. ReactiveX is a library used for developing asynchronous and event based programs using observable sequence.This observer pattern promotes more real time experiences of the mobile or web application. Even modifying a single field in the current page triggers a instant save to back end, for example Twitter follow feature enabled for some profile can be immediately reflected to other connected user and so forth.

3.3.2 jquery - librares

jQuery is a open source cross-platform javsScripy library designed to simplify the client side scripting of HTML. jQuerys is used to handle the HTML DOM traversing, event handling and Ajax interactions.Ajax enables a sleeker interface where actions can be performed on pages without reloading the whole page.

Chapter 4

Business Layer

Primary goal of the business Layer is have maximum possible modules separated logically for easy maintenance.This layer contains the business/domain logic, i.e. rules that are particular to the problem that the application has been built to handle. Complex data structure which solves the business problem are implemented in this layer. The entities related to the business problem are declared in such a way that the relationship among the entities are maintained.Business entities store data values and expose them through properties; they contain and manage business data used by an application and provide stateful programmatic access to the business data and related functionality. Business work flow is one of the major component implemented in business layer.The sequence of user task and the processing business logic to satisfy business rules differs according the scenarios and the use case. Work flow can be used to sequence and coordinate the functionality to complete them. There are three basic types of work flow style : Sequential, State machine and Data driven.

- **Sequential :** This work flow controls the sequence of activities and decides which of the process will be executed next. Although a sequential work flow can include conditional branching and looping, the path it follows is predictable.
- **State Machine :** In this style, the work flow acquires a given state and waits for events to occur before moving into another state. Consider the state-machine style if you require work flows designed for event driven scenarios, user interface page flows such as a wizard interface, or order processing systems where the steps and processes applied depend on data within the order.
- **Data Driven :** n this style, information in the document determines which activities the work flow will execute. It is appropriate for tasks such as a document approval process.

4.1 Web Server

Web Server is a platform or environment where a server-side program can be execute and hosted in Web server's port. A Web server handles the HTTP protocol. When the Web server receives an HTTP request, it responds with an HTTP response, such as sending back an HTML page. To process a request, a Web server may respond with a static HTML page or image, send a redirect, or delegate the dynamic response generation to some other program such as CGI scripts, JSPs (JavaServer Pages), servlets, ASPs (Active Server Pages), server-side JavaScripts, or some other server-side technology.

4.1.1 NGINX

NGINX is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server and a generic TCP/UDP proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.Handling high concurrency with high performance and efficiency has always been the key benefit of deploying nginx. It provides the key features necessary to conveniently offload concurrency, latency processing, SSL (secure sockets layer), static content, compression and caching, connections and requests throttling, and even HTTP media streaming from the application layer to a much more efficient edge web server layer. It also allows integrating directly with memcached/Redis or other "NoSQL" solutions, to boost performance when serving a large number of concurrent users. It has ability to handle more than 10,000 simultaneous connection. Nginx uses the event handling pattern which handles a service requests concurrently to a service handler by one or more inputs. It is a single threaded but can fork several processes to utilize multicore.

4.1.2 Apache HTTP Server

Apache HTTP Server is a one of the open source web server most commonly used in the world. Apache provides a variety of multi-processing modules (Apache calls these MPMs) that dictate how client requests are handled. Basically, this allows administrators to swap out its connection handling architecture easily. Apache servers can handle static content using its conventional file-based methods.

Apache can also process dynamic content by embedding a processor of the language in question into each of its worker instances. This allows it to execute dynamic content within the web server itself without having to rely on external components. These dynamic processors can be enabled through the use of dynamically loadable modules. Apache's ability to handle dynamic content internally means that configuration of dynamic processing tends to be simpler.

4.2 Business Process Management

Business Process Management is supported by Business Process Model and Notation (BPMN) by providing a notation that is intuitive to business user, yet able to represent complex process semantics. A business process allows you to model the business goals by describing the steps that need to be executed to achieve that goal and the order, using a flow chart. This greatly improves the visibility and agility of your business logic, results in higher-level and domain-specific representations that can be understood by business users and is easier to monitor.

4.2.1 Java Business Process Management

The core of jBPM is a light-weight, extensible workflow engine written in pure Java that allows you to execute business processes using the latest BPMN 2.0 specification. It can run in any Java environment, embedded in your application or as a service. jBPM supports adaptive and dynamic processes that require flexibility to model complex, real-life situations that cannot easily be described using a rigid process. We bring control back to the end users by allowing them to control which parts of the process should be executed, to dynamically deviate from the process, etc.

jBPM is also not just an isolated process engine. Complex business logic can be modeled as a combination of business processes with business rules and complex event processing. jBPM can be combined with the Drools project to support one unified environment that integrates these paradigms where you model your business logic as a combination of processes, rules and events.

4.3 Platform as a Service

4.3.1 Heroku

4.3.2 Google App Engine

Chapter 5

Data Layer

Data layer consist of logics to access data and handler to manage the service provided by the other applications. Centralising the Data access logics makes the application easier to configure and maintain.The frameworks are available which comes with in built data access and data sources structure to application entities mapping functionality. While requesting for the service from the other application, there is a need for the functionality which manages the process of communication with that service provider, offline support and basic mapping between data exposed from service provider and the application format. Both of these services and the data access logics are mostly abstracted from other layers as a part of security and maintenance. The data consumer interact with the abstract interface using business entities .

5.1 Cache

5.1.1 Ehcache

Ehcache is an open source in-memory cache that boosts performance by reducing the redundant operations. The redundant operations are avoided by storing the result created by that time consuming functions or frequently used data from databases in local memory which in turn improves the performance of the algorithm incorporated in business layer and simplifies scalability. It's the most widely-used Java-based cache because it's robust, proven, full-featured, and integrates with other popular libraries and frameworks. Ehcache requires no initial configuration which makes it easy to maintain on deployment. Big Memory Go Ehcache's project easily integrates with the Hibernate Object/Relational persistence and query service.Big memory allows application to store data outside JVM (off-heap store) that is in machine memory for ultra fast access. The combination of Big Memory with Ehcache grealty extends the memory available for caching.

The Ehcache API is used in the following topologies:

- **Standalone :** The cached data set is held in the application node that has no communication with other application. If standalone caching is being used where there are multiple application nodes running the same application, then the consistency between them is lost. They contain consistent values for immutable data or after the time to live on an Element has completed and the Element needs to be reloaded.
- **Distributed :** Distributed caching enables data sharing among multiple Cache-Managers and their caches in multiple JVMs. By combining the power of the Terracotta Server Array with the ease of Ehcache application-data caching,consistency in data is achieved and increases the application performance with distributed in-memory data.Distributed caching is recommended in a clustered applications.
- **Replicated :** The cached data set is held in each application node and data is copied or invalidated across the cluster without locking. Replication can be either asynchronous or synchronous, where the writing thread blocks while propagation occurs.

Many production applications are deployed in clusters of multiple instances for availability and scalability. However, without a distributed or replicated cache, application clusters exhibit a number of undesirable behaviors, such as:

- **Cache Drift :** if each application instance maintains its own cache, updates made to one cache will not appear in the other instances. This also happens to web session data. A distributed or replicated cache ensures that all of the cache instances are kept in sync with each other.
- **Database Bottlenecks :**In a single-instance application, a cache effectively shields a database from the overhead of redundant queries. However, in a distributed application environment, each instance much load and keep its own cache fresh. The overhead of loading and refreshing multiple caches leads to database bottlenecks as more application instances are added. A distributed or replicated cache eliminates the per-instance overhead of loading and refreshing multiple caches from a database.

5.2 Message Queue

Message queue provides asynchronous communication between multiple processes or multiple threads within a process that places the message or request in a queue without processing it immediately. It is used in the application where the incoming request are to be processed in specific order. Considering a application with the web service which receives many requests every second where no request can be lost and all the requests needs to be processed which is time consuming. This scenario requires message queue which would manage all the request without losing them.In the scenario where the application is decoupled and the multiple components which is decoupled exchange information by sending messages between processes.

5.2.1 RabbitMQ

RabbitMQ is a open source multiple protocol messaging server written in Erlang which implements Advanced Messaging Queuing Protocol (AMQP) and other services such as the Amazon Simple Queue Service (SQS). RabbitMQ is used by Instagram , Indeed.com ,Mozilla etc. RabbitMQ offers a better approach to hard coding information exchange patterns. When you use the messaging service, you create a producer that writes a message to a broker, which is responsible for accepting messages and routing them to consumer applications. RabbitMQ brokers consist of three components: exchanges, bindings and queues. Exchanges receive messages and distribute them to queues where consumer applications can read them.

5.2.2 Kafka

Apache Kafka is a fast, scalable, durable and fault-tolerant open source publish-subscribe messaging system. Kafka is often used in place Java message service and Advanced Messaging Queuing Protocol (AMQP) because of its higher throughput, reliablitiy and replication.Apache storm(event processor) and Apache HBase(Hadoop Distributed Filesystem) both works very well in combination with kafka.

5.3 Object Relational Mapper

Object Relational Mapper(ORM) is mechanism of mapping the objects in the application to the relational objects of the data store. ORM provides a bunch of services for handling crud operations on applications database.ORM tools provide an object oriented query language which allows application developers to focus on the object model and not to have to be concerned with the database structure or SQL semantics. The ORM tool itself will translate the query language into the appropriate syntax for the database.The entire transaction can either be committed or rolled back. Multiple transactions can be active in memory in the same time, and each transactions changes are isolated form on another. It helps developer to avoid writing complex and tedious sql statement and eliminates the usage of JDBC APIs for result set or data handling.

5.3.1 Hibernate

Hibernate is an open source jave persistence framework implementated based on Java Persistent API (JPA). It provides HQL (Hibernate Query Language), which is compatible with any database server and the hibernate internally converts HQL to underlying database query format and executes it. In ORM the table is mapped to java object allowing the use of OOP concepts like inheritance , encapsulation, etc. Hibernate provides caching mechanism which reduces number of database hit. Hibernate provides lazy fetching which helps to fetch required data contained in the object.

Chapter 6

Data Store

Data store component can be considered as repository to store and manage the collection of information used in application. Generally data store can be classified as database in which data is stored in organised manner so that it can be easily accessed, managed and updated. For each application the database is designed according to the business logics and entities such that the data can be interpreted by components for the future use. The database management system(DBMS) is a system software used for creating and managing the databases. DBMS serves as an interface between database and the user or application programs, ensuring that data is consistently organized remains easily accessible.

Database can be categorised into different types on basis of the structure of the stored data and the functionality.

6.1 Relational Database

Relation database organises data into one or more tables which is formally described and organised according to the relational model . Each row in the table which is also called as record is a set of values related with each other in some way and a unique key identifying each row. Its focus is to implement simple CRUD - create, Read, update and Delete functionality. Each row represents an object and information about object.A relationship is established among the tables based on the interaction between them which is derived from business logics of the application. Currently Mysql , PostgreSql,SQLite and Oracle are the popular RDBMS used for applications. A connection is established between the application server and the database server using drivers. All the dbms has its own driver which provides API that defines how a client may access a database. Unlike database that exit in server side, SQLite is a in-memory embedded software used in client storage in the apllication such as web browser, mobile client etc.SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file. The database file format is cross-platform. These features make SQLite a popular choice as an Application File Format. Basically SQLite strive to provide local data storage for individual applications and device. But Mysql , PostgreSql and Oracle are Client/server SQL database engines aims at providing a shared repository of enterprise data.

- **Connection Establishment:** Mysql , PostgreSql and Oracle are separate process running in the machine. Back end of the applications establishes connectivity with these DBMS using dedicated drivers API. But SQLite library is imported into the client side of an application and its functionality is called through function calls which is reduce latency in database access as SQLite doesn't have separate server process. Because function call within a single process is more efficient than inter-process communication.
- **Access Control:** To use databases in client-server systems, the connection establishment requires a user authentication and user based authorization service.But SQLite doesn't require any authorization or authentication.Access control of SQLite is based on the file system permissions given to the database file itself.
- **Concurrent Access :** As client server database model focus on shared memory, it allows multiple user making it highly concurrent. For example MySQL with InnoDB storage engine does row level locking to support simultaneous write access by multiple sessions. MySQL with MyISAM storage engine does table level locking allowing single session to update tables at a time. But this can be alter to multi user by changing the concurrent_insert system variable. SQLite design grants lock to the entire databases file during writing allowing concurrent read and sequential writes.

6.1.1 Usage

6.2 Mobile Database

Mobile application can have a centralized database which is connected to mobile over network and a database actually stored in the mobile. Mobile device prefer database replication which means frequent retrieval of data from a centralised database and store it in the local database in the mobile.But bandwidth usage must be taken into consideration. Mobile client database contains a subset of data stored in the centralised database. Data synchronization is important to maintain the integrity between mobile server and mobile client. In some application, Queues are used to manage the information for synchronization.Most of the case, synchronization is asynchronous and the change propagation is not immediate. The best known mobile database for android and iOS is SQLite. But SQLite use complex schemas for data storage and access as it is relational database which impacts the user experience. Since SQLite doesn't have a inbuilt sync functionality, the sync code is self implemented which is considered to time consuming.

6.2.1 Couchbase Mobile

Couchbase mobile is the first NoSQL JSON database that offers full flexibility of NoSQL to mobile.It is designed to provide fast and consistent access to data offline or online. Couchbase mobile consist of three components:i)

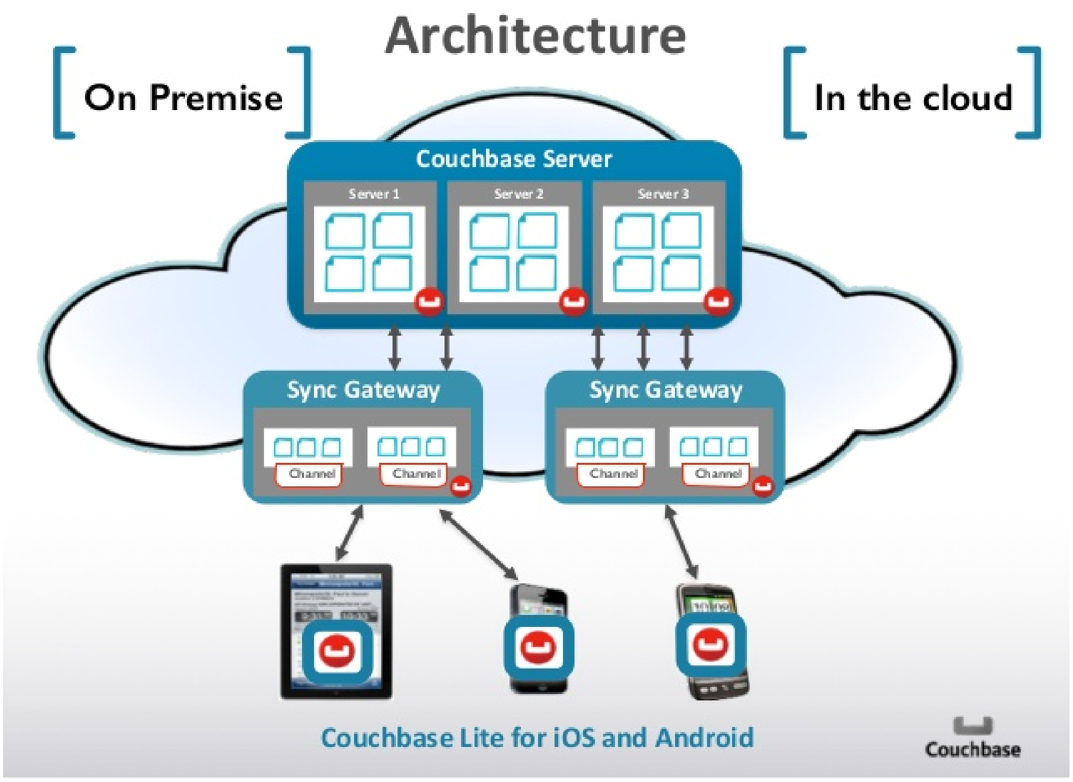


Figure 6.1: Couchbase Mobile Architecture

- **Couchbase Lite:** Couchbase Lite is an lightweight embedded full featured JSON document based and schemaless database on the device. It has APIs for Android,iOS and Rest along with integration support in popular cross-platform tools such as Phonegap and Xamarin. It includes features like peer-to-peer replication along with authorization by connecting with other mobile device running Couchbase Lite as its local storage and sync data with them after determining the document is replicated for that user.

- **Couchbase Sync Gateway:** This component handles a two-way change synchronization between JSON store in Couchbase Server and Couchbase Lite.It provides feature like routing required data to users avoid frequent network API calls, plug-in authentication, change validation and access control.
- **Couchbase Server:** The NoSQL database that resides in the backend with elastic scalability and consistent high performance.

6.2.2 Realm

Realm is a open source library launched in 2014 which can be integrated with the mobile application to store and query data. Realm introduced many features which made it appear as a replacement for SQLite.

- In this database , data is exposed as objects directly incorporating object relationships avoiding the need for ORM. At the same time, it remains extremely memory efficient.
- It includes full ACID transactions default and an object schema is created directly driving from object definition.
- Unlike SQLite , Realm database are safe across threads making it highly concurrent when there is a burst of asynchronous task .

6.3 Key - Value Database

Key-Value database is considered to be a simplest way storing data in the form of key-value pair and retrieving data based on key value. An advanced form of key-value store introduces the sorting of keys which enables an ordered processing of keys. Key-value database is also known as NoSQL databases. Key-Value store works differently when compared to relational database and it addresses several issues which relational database didn't address. Currently Oracle NoSQL , MongoDB and Redis are popularly used NoSQL database.

- **Dynamic Schemas :** Relational databases expects the schemas to be pre-defined which is poorly suited for agile development approaches as the schemas changes as the feature evolves.NoSQL allows insertion of data without predefined schema.This feature makes the applications changes easy in real time which promotes reliable code integration.
- **Auto-Sharding :** It is supported by NoSQL databases which allows them to natively and automatically spread data across an arbitrary number of servers, without requiring the application to even be aware of composition of server

pool.Using cloud computing makes sharding way more easier providing the same processing and storage capabilities as a single high-end server for a fraction of the price. Relational database is hosted in a single server which is vulnerable to single point of failure. Even though many database server are integrated programatically, the transactional integrity are eliminated by employing manual sharding.

- **Integrated Caching :** Many frameworks or Tools provides caching for SQL database systems which can improve read performance , but they do not improve write performance. NoSql offers completely managed integrated in-memory database management layer for the workloads demanding the highest throughput and lowest latency.

6.4 Cloud Storage

Cloud storage is a service where data is remotely maintained, managed and backed up. This service enables the user store data online which can be accessed through remote servers and cuts down on the space you use on the application server computer. It handles sharding and replications automatically which provides highly stable database that handles any type of application load. Mostly the cloud dataStore are schemaless database which allows flexible to changes made in application entities as it evloves. Amazon Web Services(AWS) and Google Cloud Datastore are the mostly used cloud platform in recent days. Amazon web service provides cost effective object storage for different varieties of cloud-based database service including both relational and NoSQL databases. Amazon Relational Database either run Mysql ,Oracle or SQL Server, while Amazon SimpleDB offers schemaless database which is suitable for smaller workload. Amazon DynamoDB is a NoSQL database backed by a solid state drive which is considered to be AWS's fastest growing service. Google Cloud DataStore is a NoSQL document database built for automatic scaling and high performance.It also handles automatic sharding and replication, providing you with a highly available and durable database that scales automatically to handle application load.

Chapter 7

Conclusion

In past three years, there was drastic outbreak of off shelf tools which is pluggable and reduced the implements done for software products. In this ever changing world of software development , it is extremely important to keep up with current technologies, methodologies and trends. This document has given an overview of software stack used in industrial environments for developing a software products and the frameworks that are widely used in current trend. Selection of the tools is key in developing a highly efficient software with good performance.

Building a great software product depend on the development and management process. Agile method is one of the methods which really helps to standardize the management process which in turn impacts the development process. At first validate the problem that has to be solve. After getting clear understanding of the problem, build a prototype on the problem. This greatly helps in starting of development process. After validating the prototype , optimize and refine it. Select the most suitable technologies and framework which would help in building a minimum viable product for first iteration. Design and develop the minimum viable product. Follow iteration based development where development of group of features in each iterations.This gives a good result where the software is well tested and optimized to great extent.

Appendix A

Appendix A name

Here is the first appendix