
Survey On Application Software

- Industrial Insights -

Project Report
Aarthi Babu
Raja Gunasekaran

University Of Northern British Columbia
Computer Science

Contents

1	Introduction	3
1.1	Command-Line Application	3
1.2	Desktop Application	4
1.3	Web Application	4
1.4	Mobile Application	4
2	Software Application Architecture	7
2.1	Basic Architecture Layer	7
2.2	Requirement-Based Components	9
2.2.1	Issues and Add-on - Presentation Layer	9
3	Web Application	11
4	Mobile Application	13
5	Conclusion	15
A	Appendix A name	17

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.

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.

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 their 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

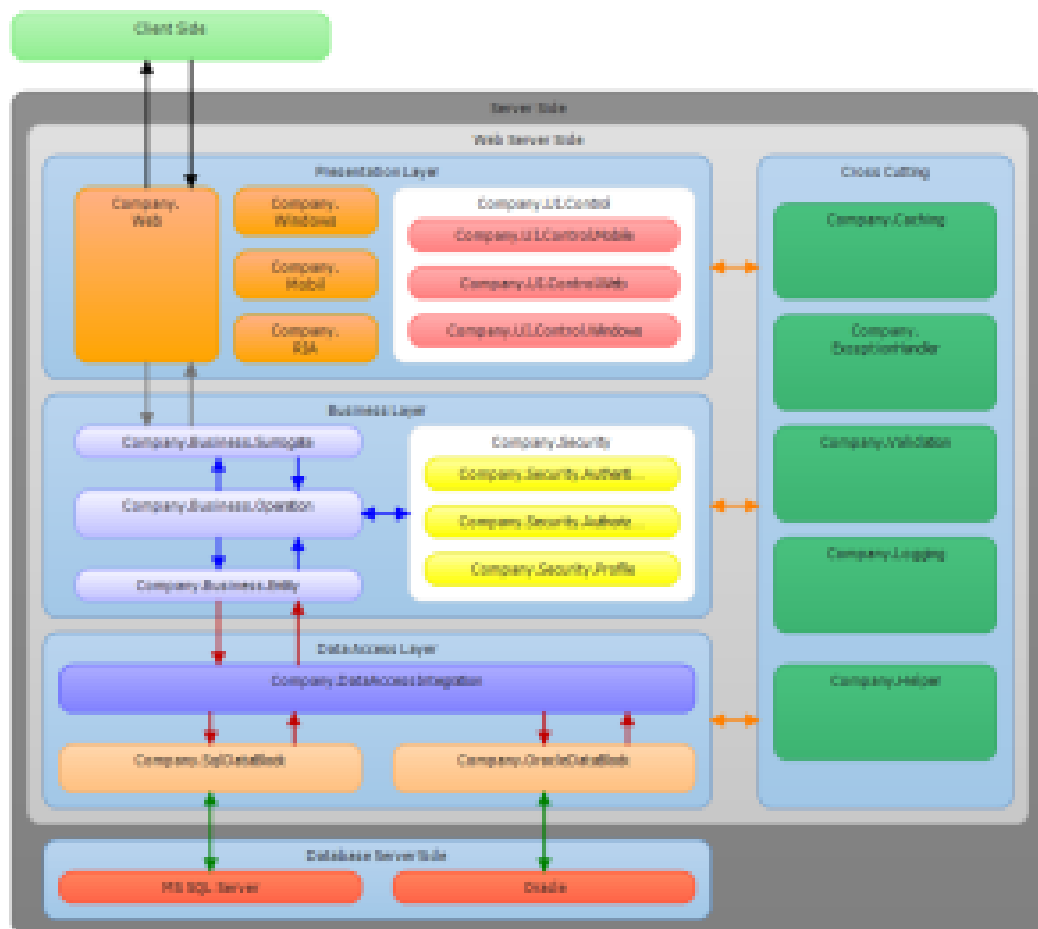


Figure 2.1: Software Application Architecture

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

- **Caching :**

2.2.3 Issues and Add-on - Data Layer

- **Caching :**

2.2.4 Issues and Add-on - Service Layer

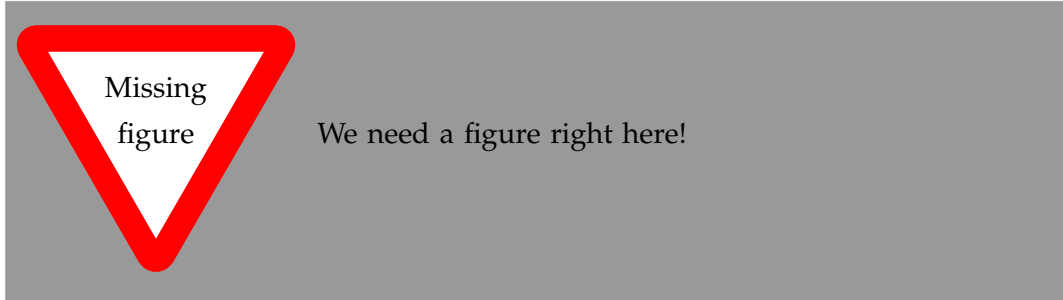
- **Caching :**

Chapter 3

Web Application

Here is chapter 2. If you want to leearn more about $\text{\LaTeX}2_{\epsilon}$, have a look at [Madsen2010], [Oetiker2010] and [Mittelbach2005].

I think this word is misspelled

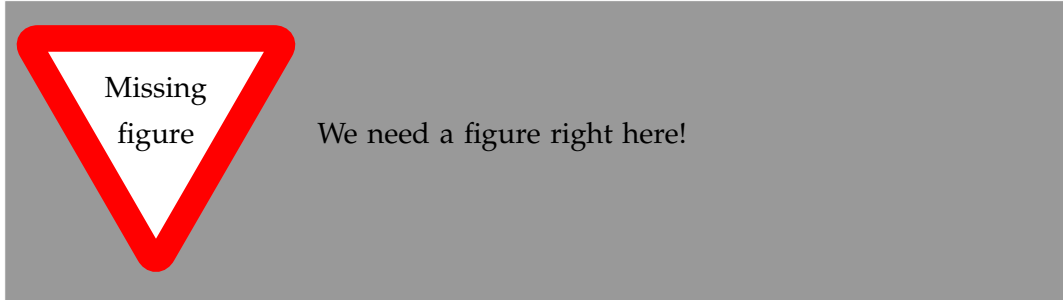


Chapter 4

Mobile Application

Here is chapter 2. If you want to learn more about $\text{\LaTeX}2_{\epsilon}$, have a look at [Madsen2010], [Oetiker2010] and [Mittelbach2005].

I think this word is misspelled



Chapter 5

Conclusion

In case you have questions, comments, suggestions or have found a bug, please do not hesitate to contact me. You can find my contact details below.

Jesper Kjær Nielsen
jkn@es.aau.dk
<http://kom.aau.dk/~jkn>
Fredrik Bajers Vej 7
9220 Aalborg Ø

Appendix A

Appendix A name

Here is the first appendix