

I58.258 Web Development

Systems Analysis & Design for Web Applications and Websites

School of Mathematics & Computational Sciences (SMCS)
Massey University
(AKLI, DISD & MTUI)

Revised: 2022-07-20

Topic Learning Objectives

- At the end of this topic, you should be able to:
 - 1. Describe some of the techniques used in analysing web application system requirements*
 - 2. Use some notation from the UML related to web applications*
 - 3. Apply some aspects of the processes involved in the development lifecycle of web applications*
 - 4. Apply some common design patterns to the structure of web pages*

Topic Outline

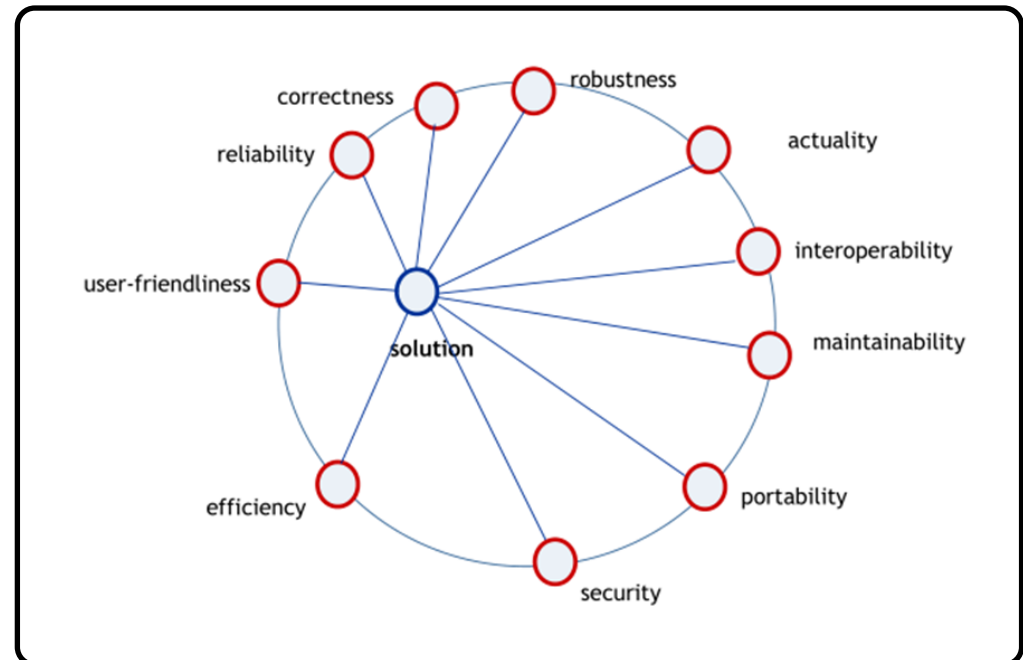
1. What's different about Web-based Systems - *Trade-offs in Web Application Design*;
2. Software Development Life Cycles (SDLC's);
3. The UML & Unified Process;
4. Modelling Requirements
5. Analysis Tools: *Domain Models, Use Cases, and Storyboards*;
6. Further Use cases;
7. From Analysis to Design;
8. Webflow Design;
9. Design Patterns for Web Page Structures

Section I: Introduction - What's different about Web-based Systems?

- Special kind of user interface;
- Large number of anonymous users;
- Data communications issues (speed, concurrency);
- *Request-response* model of iteration; and
- Client may be less rich than desktop.

Trade-off Circle for Web Application Design - I

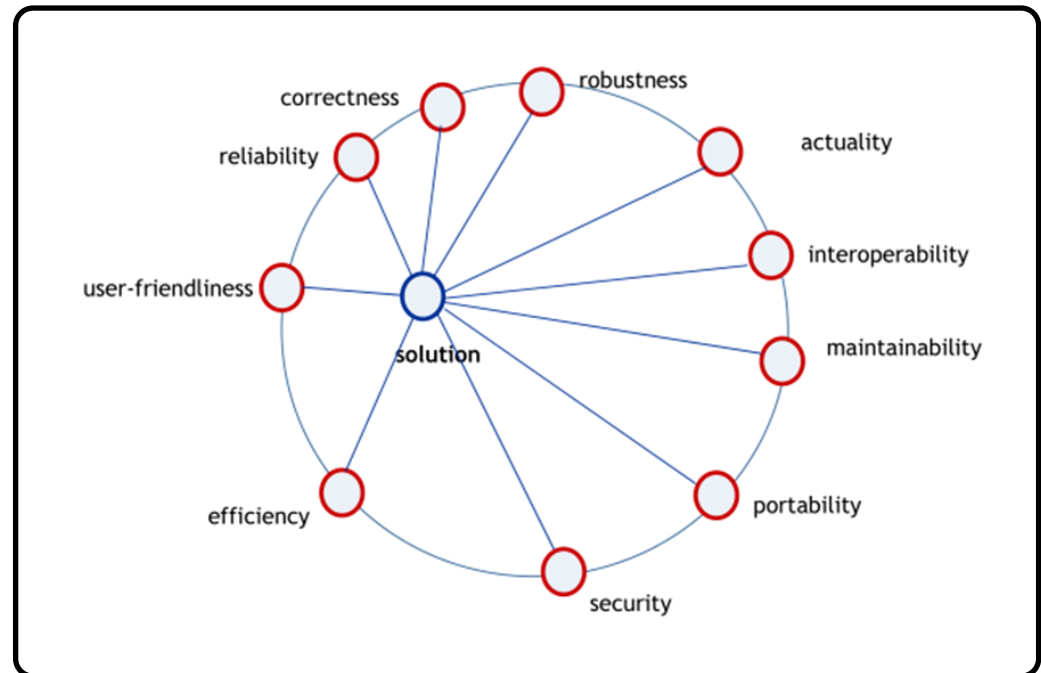
- External qualities are visible to the user and includes the following ones:
 - **correctness**: *a web application is functionally correct if it behaves according to the specification of the application*
 - **reliability**: *the probability that the software will operate as expected, occurring software errors are not serious*
 - **robustness**: *software behaves reasonably even in circumstances that were not anticipated in the requirements specification*



- **actuality:** *actuality of content must be guaranteed*
- **user-friendliness:** *easy to use by human (novice / experts)*
- **efficiency:** *economical handling of resources (time, storage space)*
- **security:** *system is protected from unauthorized access.*

Trade-off Circle for Web Application Design - II

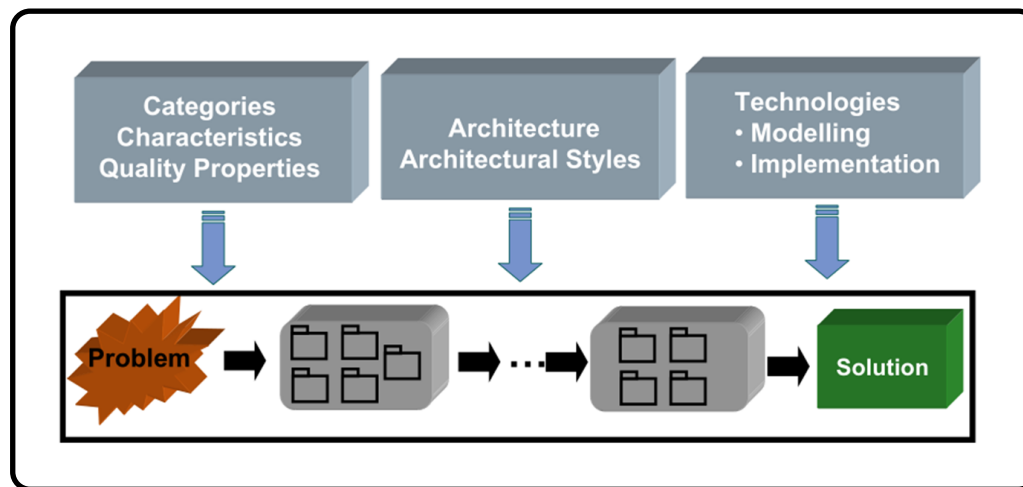
- Internal qualities are concerning and visible to the developer:
 - **portability**: a web application is portable if it can run in different environments
 - **interoperability**: refers to the ability of the web application to coexist and cooperate with other systems
 - **maintainability**: ability to modify a web application after it has been deployed, to correct errors or extend the web application



(End Section I

Section 2: Software Development Life Cycles (SDLC's)

Support for Web Application Development Process



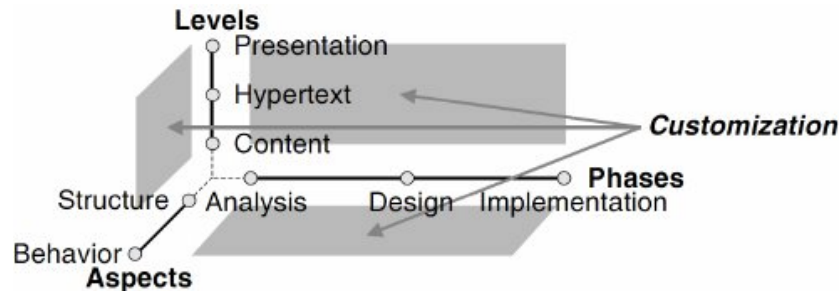
Model-based Web development

- The development of Web applications requires a methodologically sound engineering approach.
- Model-driven Web Engineering (MDWE) is a still emerging field aiming at

providing sound model-based solutions for building Web applications that try to separate the abstract design (PIM) from the concrete technological platforms (PSMs).

- However, current MDWE approaches cannot provide solutions for all kinds of the requirements against a software system but the directions are clear.

Design dimensions of Web applications



(Lajos KollárAttila and AdamkóAttila Adamkó (2011),
Interoperability issues of
MDWE methodologies;
STUDIA UNIV. BABES,–
BOLYAI, INFORMATICA,
Volume LVI, Number 3, 2011:
pp.71 - 76)

- Most of the Web app development methodologies propose the **separation of different views (i.e., models)** of the application into 3 levels:
 - *structural (or content),*
 - *navigational (or hypertext) and*
 - *presentational models.*
- Despite the separation, the **levels should be interconnected** in order

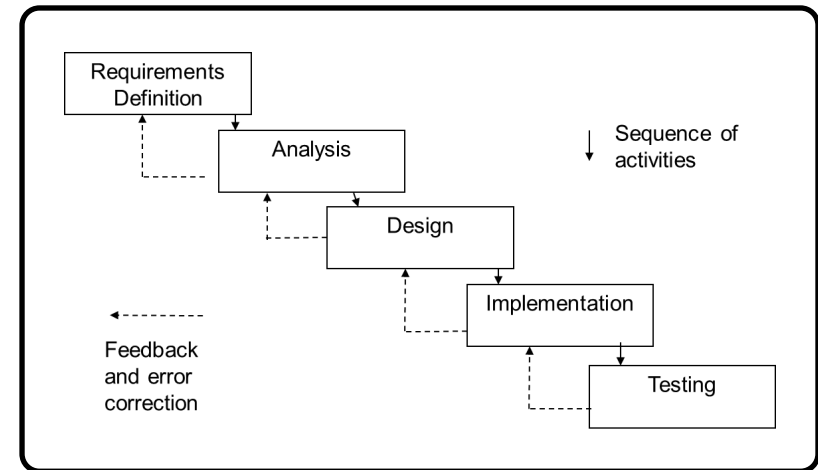
to be able to capture the semantics behind the elements of the different models:

- *e.g., the navigational objects are based on certain elements of the content model.*
- Both **structure** and **behaviour** need to be modeled using a uniform notation that has to cope with the specific characteristic of each of the levels.

The Waterfall Model: *Limitations*

Limitations

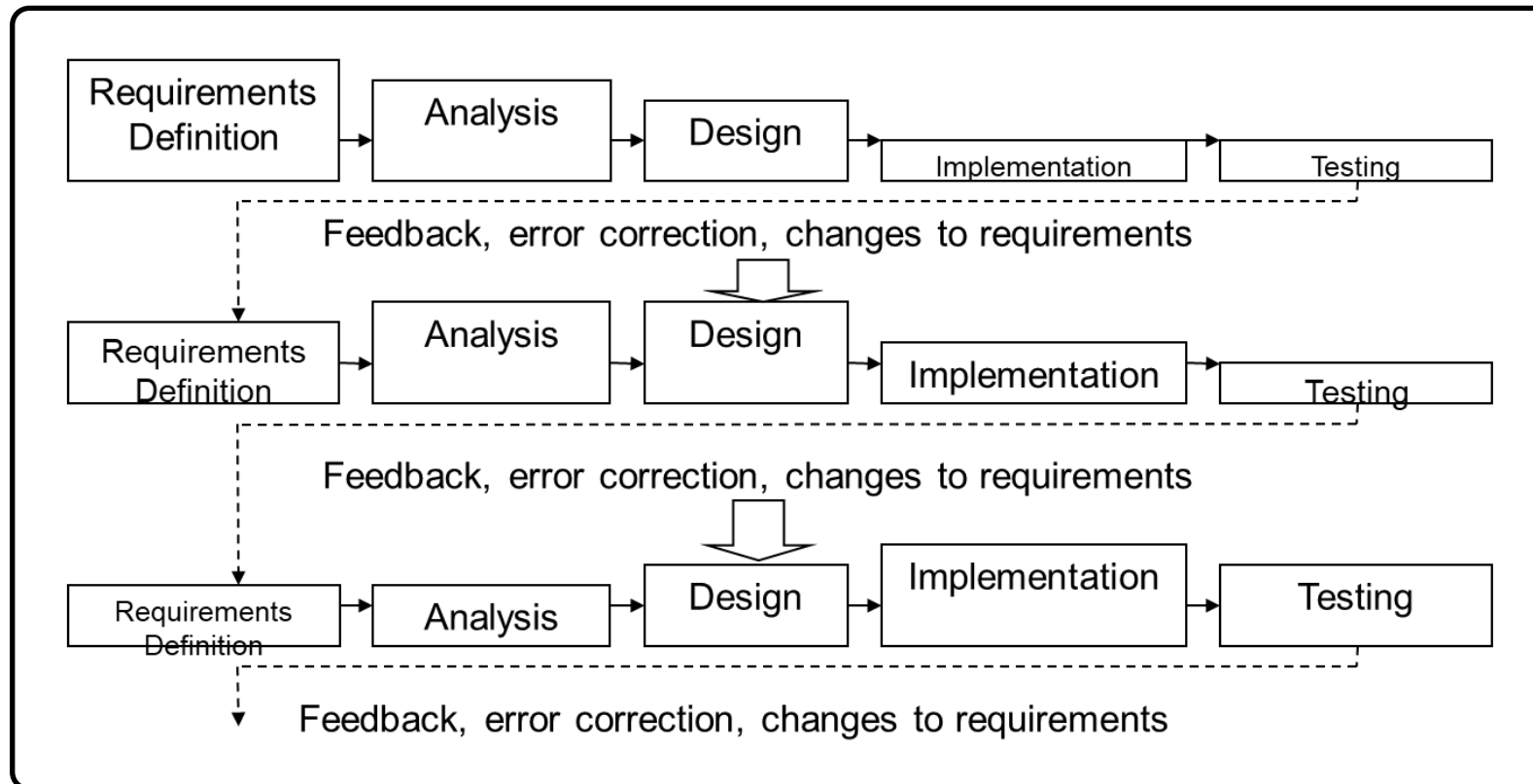
1. Clients may not know what their requirements are before they see working software and so change their requirements, leading to redesign, redevelopment, and retesting, and increased costs. (*Parnas and Clements, 1986*)
2. Designers may not be aware of future difficulties when designing a new software product or feature - revision of design may be necessary. (*McConnell, 2004*)
3. In practice, it is difficult to sustain a strict separation between systems analysis and programming. (*Ensmenger, 2010*)



The Waterfall Model: *An early approach to developing software*

The Waterfall Model: *An early approach to developing software*

The Iterative Model



Better for dynamic systems

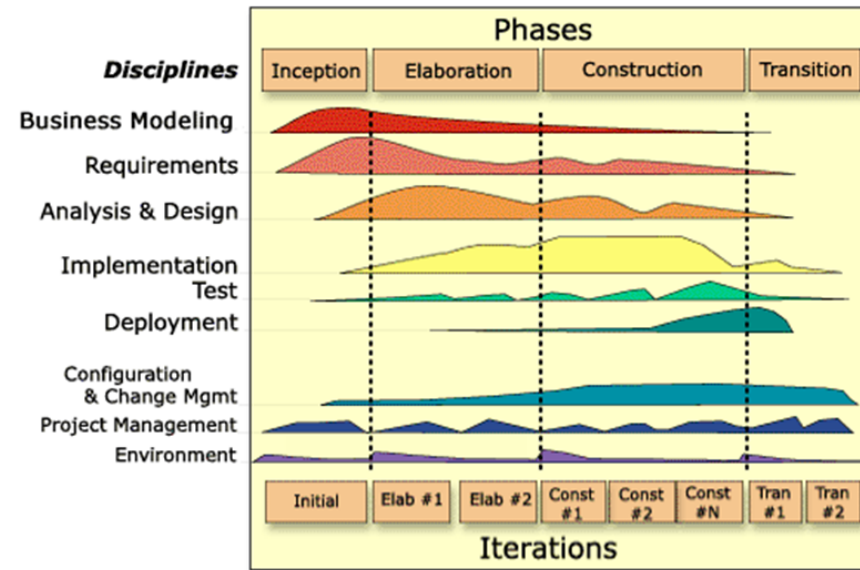
The Unified Modelling Language (UML)

- A standard set of diagrams for modelling object oriented software development;
- We do not have to use the entire language for every type of system; and
- Some 'agile' methodologies (e.g. Iconix) use a small subset of the UML.



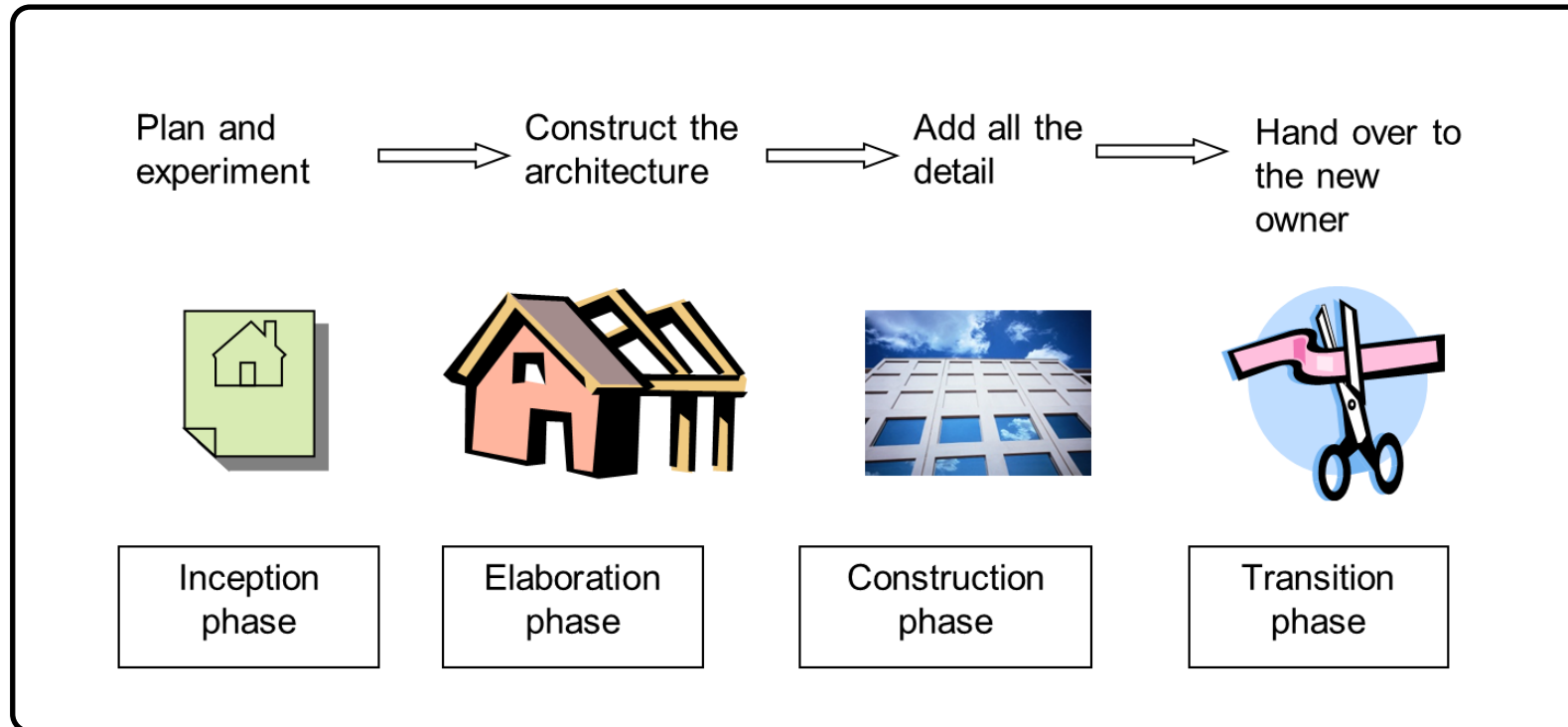
The Unified Process (UP)

- 4 phases
- 9 disciplines
- Any number of iterations (at least 4!)
- Each iteration has a milestone (deliverable)



A standard process for developing object-oriented software

The Unified Process as a building project



Analysis vs Design

- **Analysis** is:

- *defining the problem domain and*
- *specifying how we anticipate the system will be used from the user perspective*
- *technology agnostic*

- **Design** is:

- *how we plan the solution*
- *technology aware*

Design Detail

- In an **iterative process**, the transition from analysis to design is a gentle one
 - *Not like going over the waterfall*
- **Design** starts off at a high level and becomes more detailed
- To **successfully design** you need to understand the *technology of the solution*

(End: Section 2)

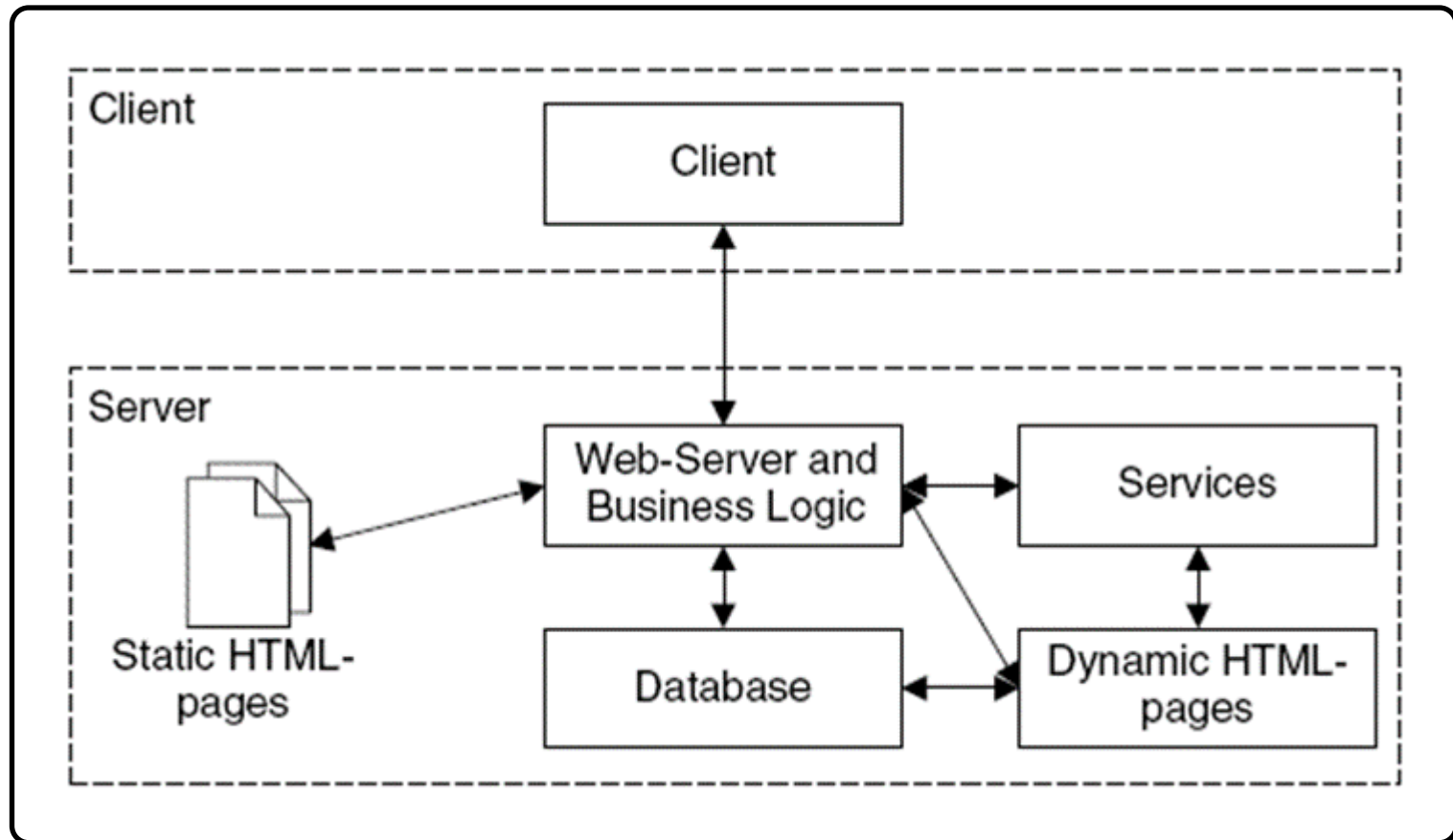
Section 3: Web Application Design

Aspects of Design

- We will look at some architectural aspects of design from the server perspective
- We will also look at some aspects of design from the client perspective
- These are design patterns:
 - *Commonly used solutions to the problems of designing web applications*

Aspects of Design: Client-Server Architecture

- An

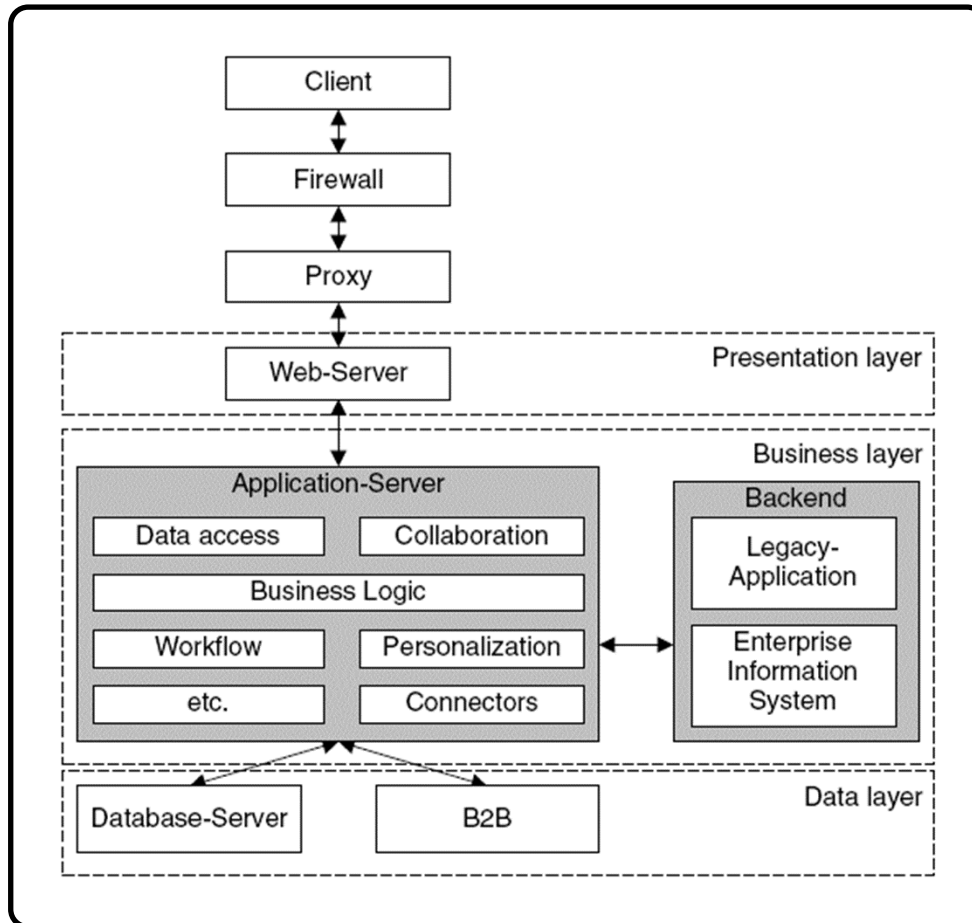


important feature of designing web based applications is allocating roles to

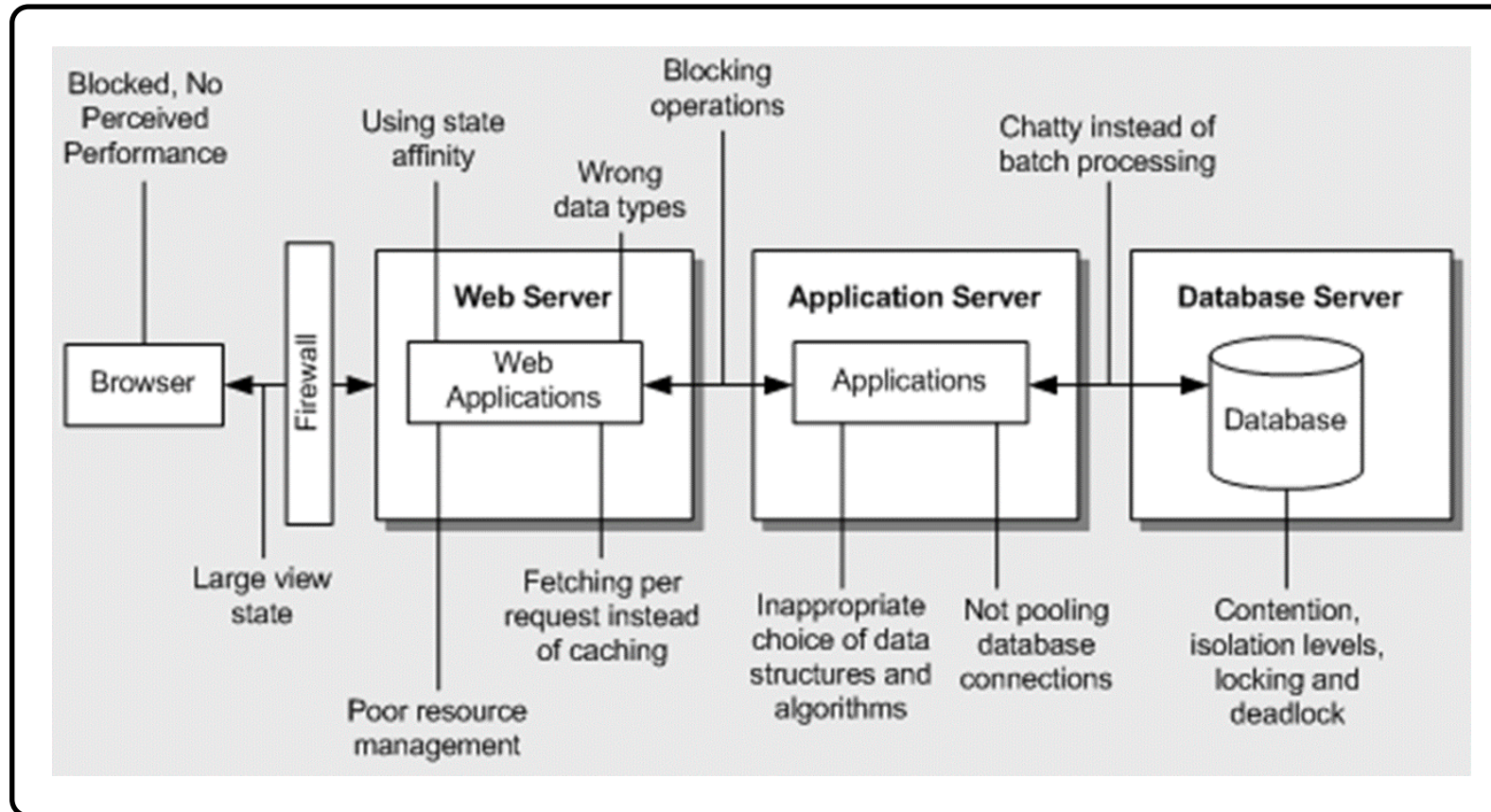
the client and the server

- How 'thin' is the client?
- Should we use the browser's ability to, for example, run JavaScript, Java Applets, Flash etc?
- What if the user has a mobile phone?

The 3-Tier Web Application Architecture



Traditional Challenges of the 3-Tier Web App Architecture



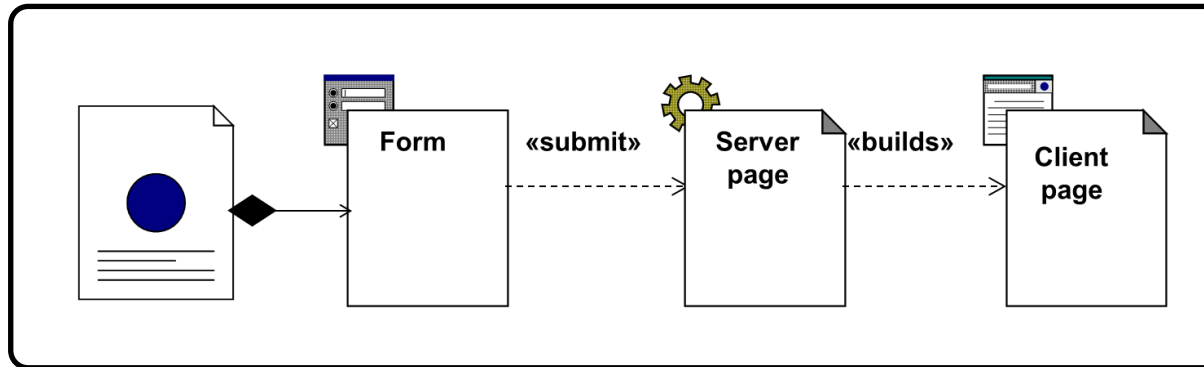
(End: Section 3)

Section 4: Website Content Design

Static and Dynamic Content

- Your design has to take into account how much of the application will be static and how much dynamic
- How much can be just **HyperText Markup Language (HTML)** documents that are served to every client?
- How much will have to be generated on the fly for specific clients?

Server Pages and Client Pages - I

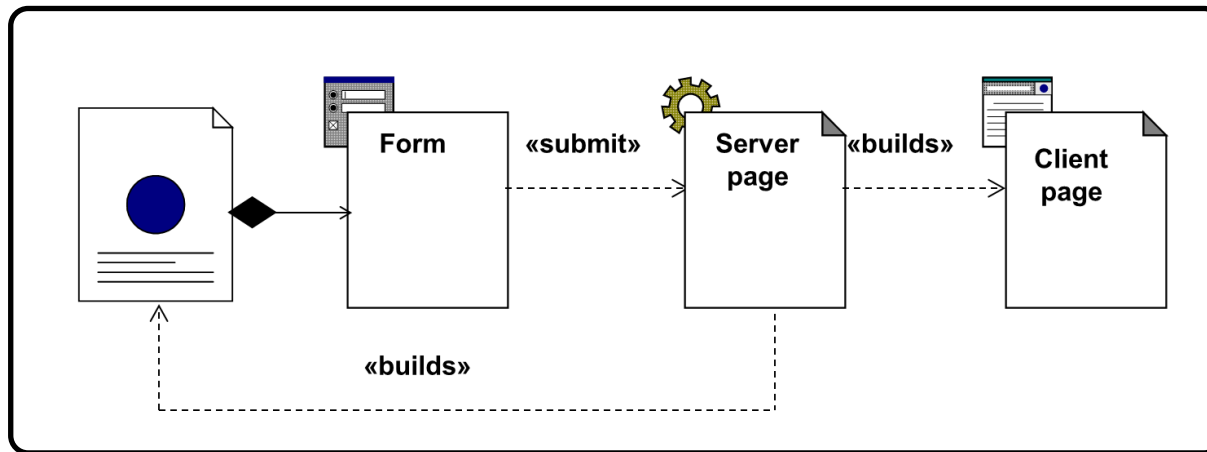


- **Server pages** are a technology for generating dynamic web pages on the server before sending them to the client as HTML:
 - *They are programs that run on the server*
 - *They can be written using one of many server-side programming languages, e.g., Java, Python and recently JavaScript*
- **Client pages** are Web pages that either present information to the user or get data from the user.

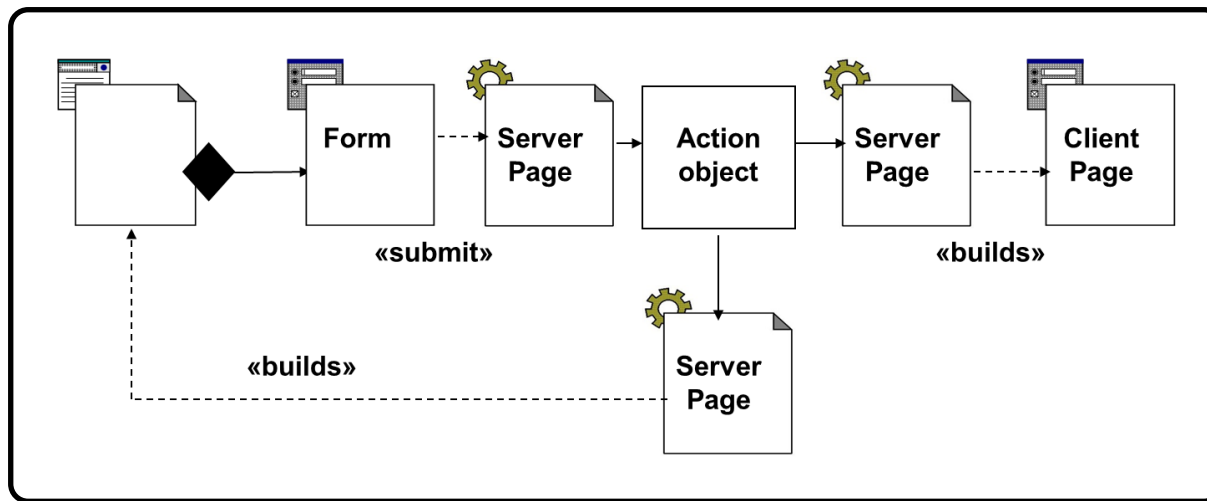
- *In the diagram, a client page has a Web form embedded in it to collect data from the user. Also, a server page runs to build a client page that contains information to be presented to the user.*

Server Pages and Client Pages - II

Dynamic Form Generation

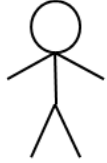


Specialized Server Pages



Dynamic Webflow UML Sequence Diagram

New User



Start Page



Use case controller

Action Object



Page Builder



Client Page Form

Select option

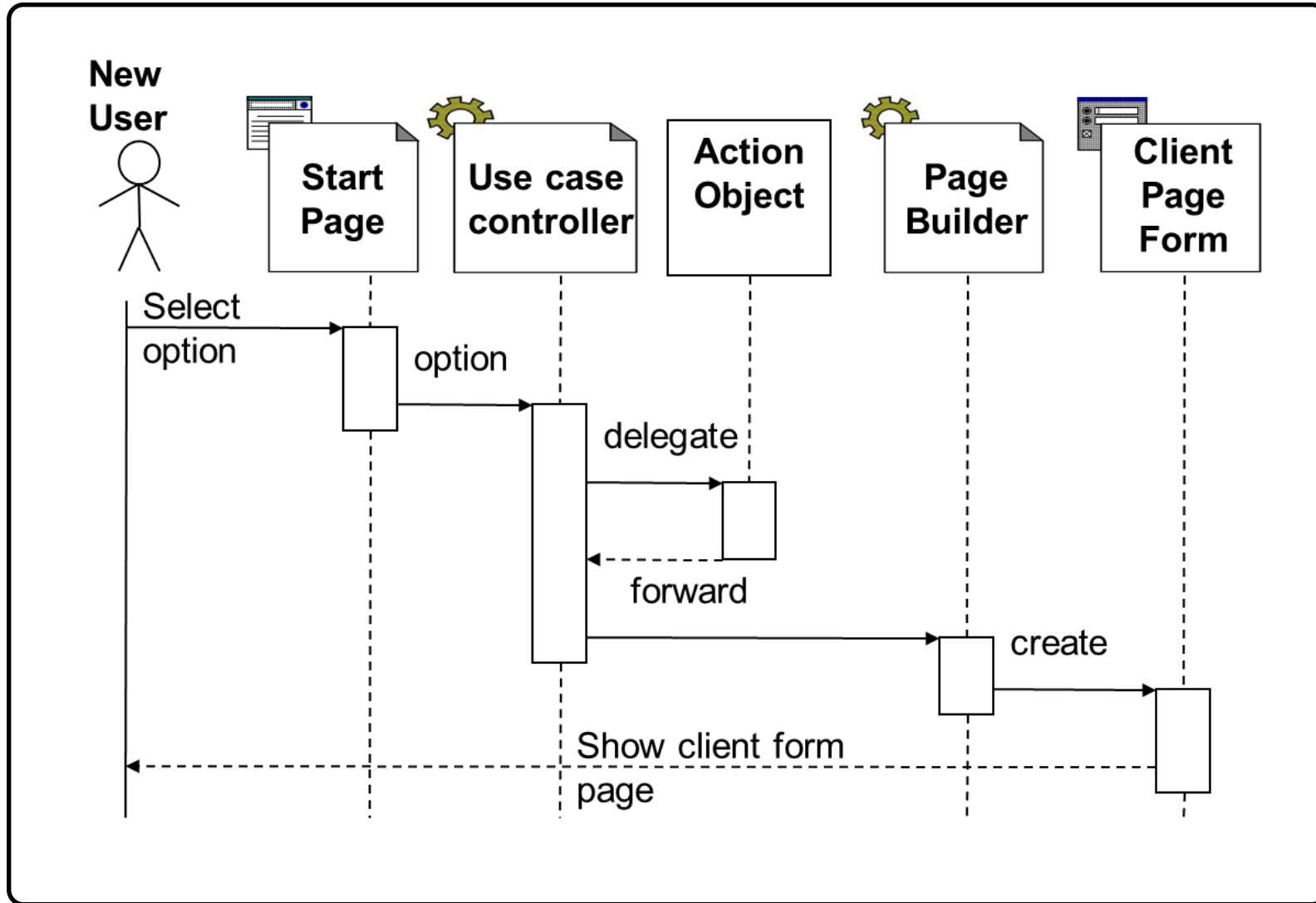
option

delegate

forward

create

Show client form page

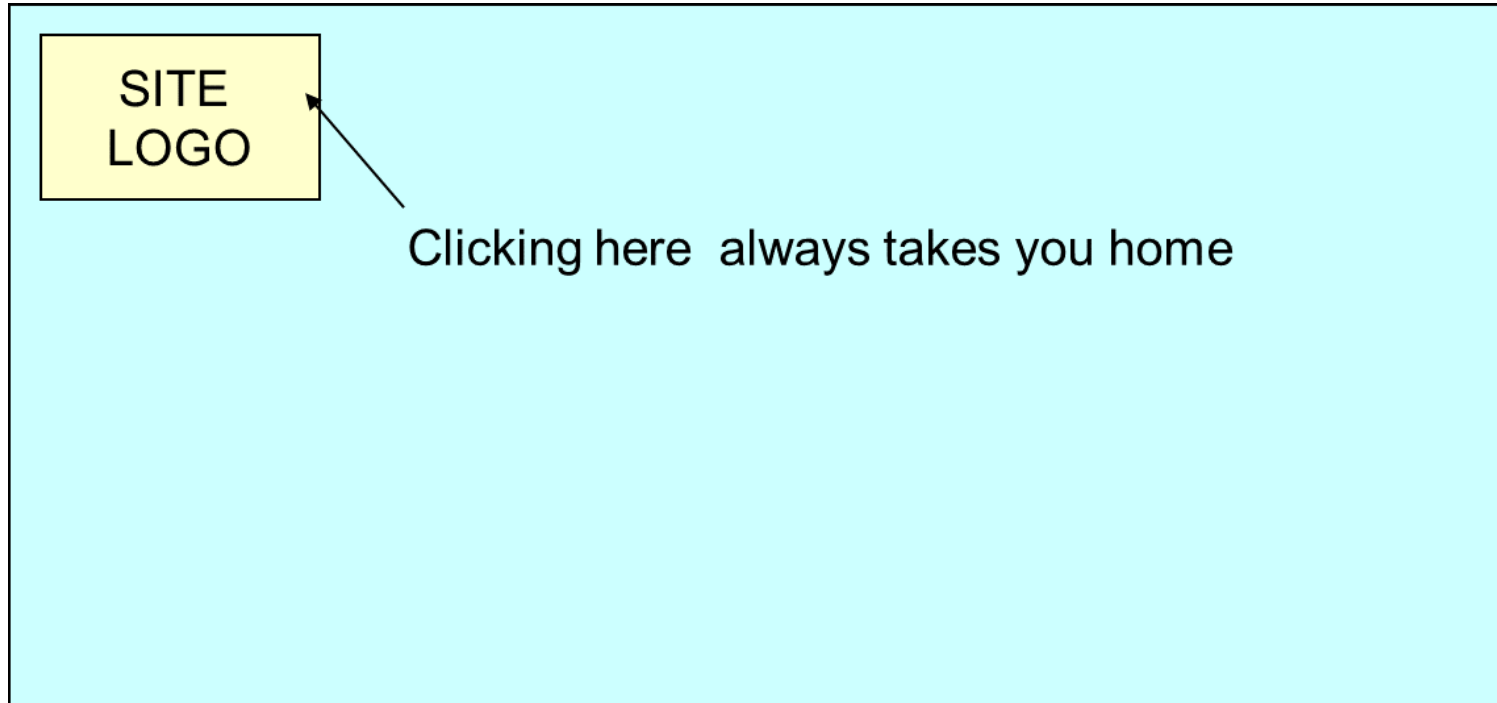


Design Patterns for Web Page Structure

- We will now ,look at some web design patterns that help us to structure our applications, which are as follows:

- 1. Site logo at top left*
- 2. Navigation bar*
- 3. Breadcrumbs*
- 4. Three region layout*
- 5. Home page*
- 6. Site map*
- 7. Store content in the database*

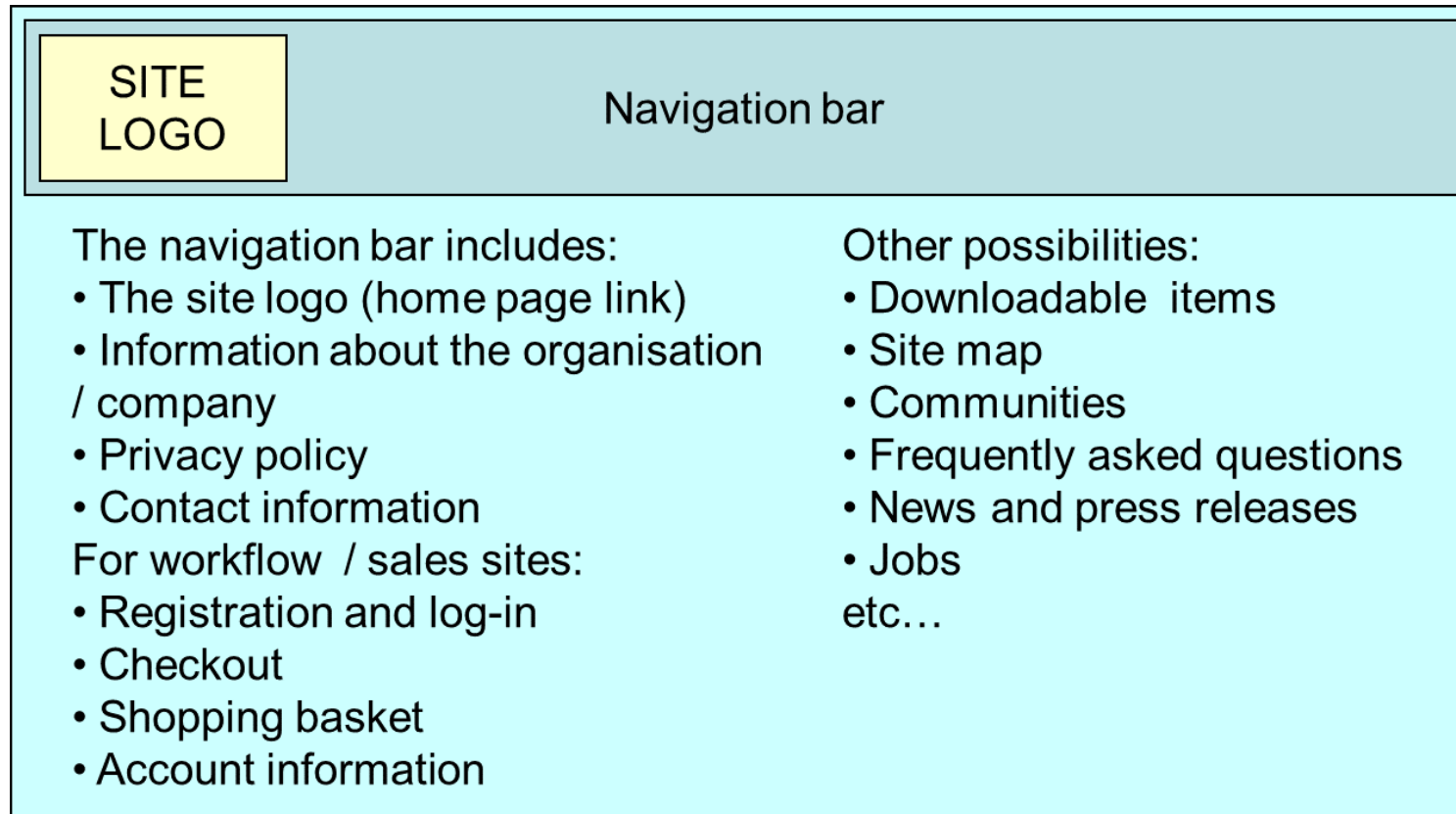
Website Logo at the Top Left



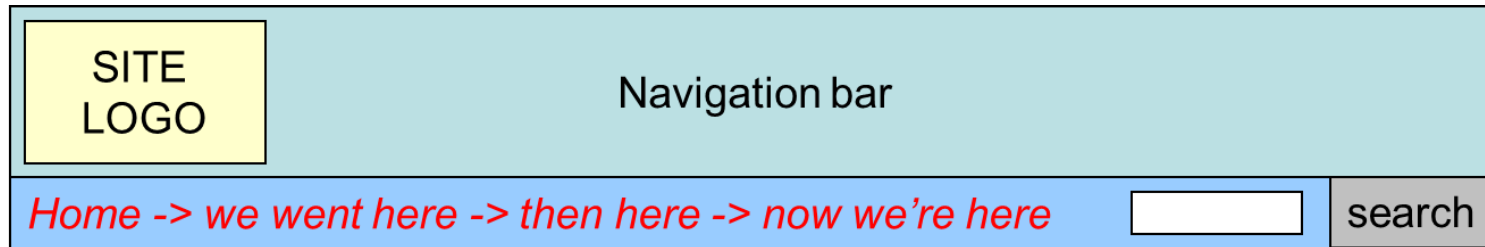
Use Cases

- Our use cases will be starting points for user navigation
- They will frequently appear in a navigation bar across the top of the page
- The left hand side can be used for service navigation (i.e. what is inside the current use case)

The Navigation Bar

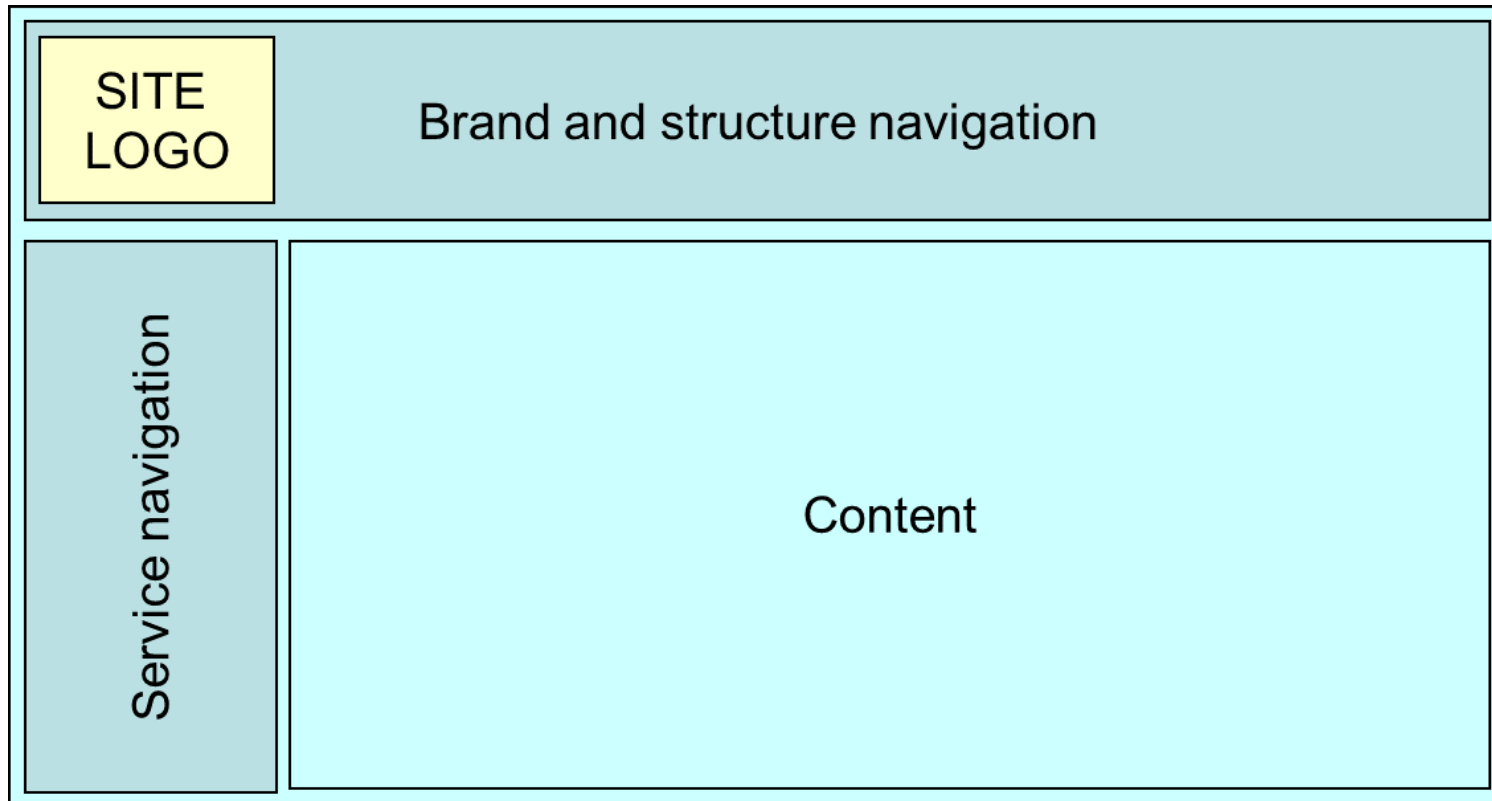


Breadcrumbs



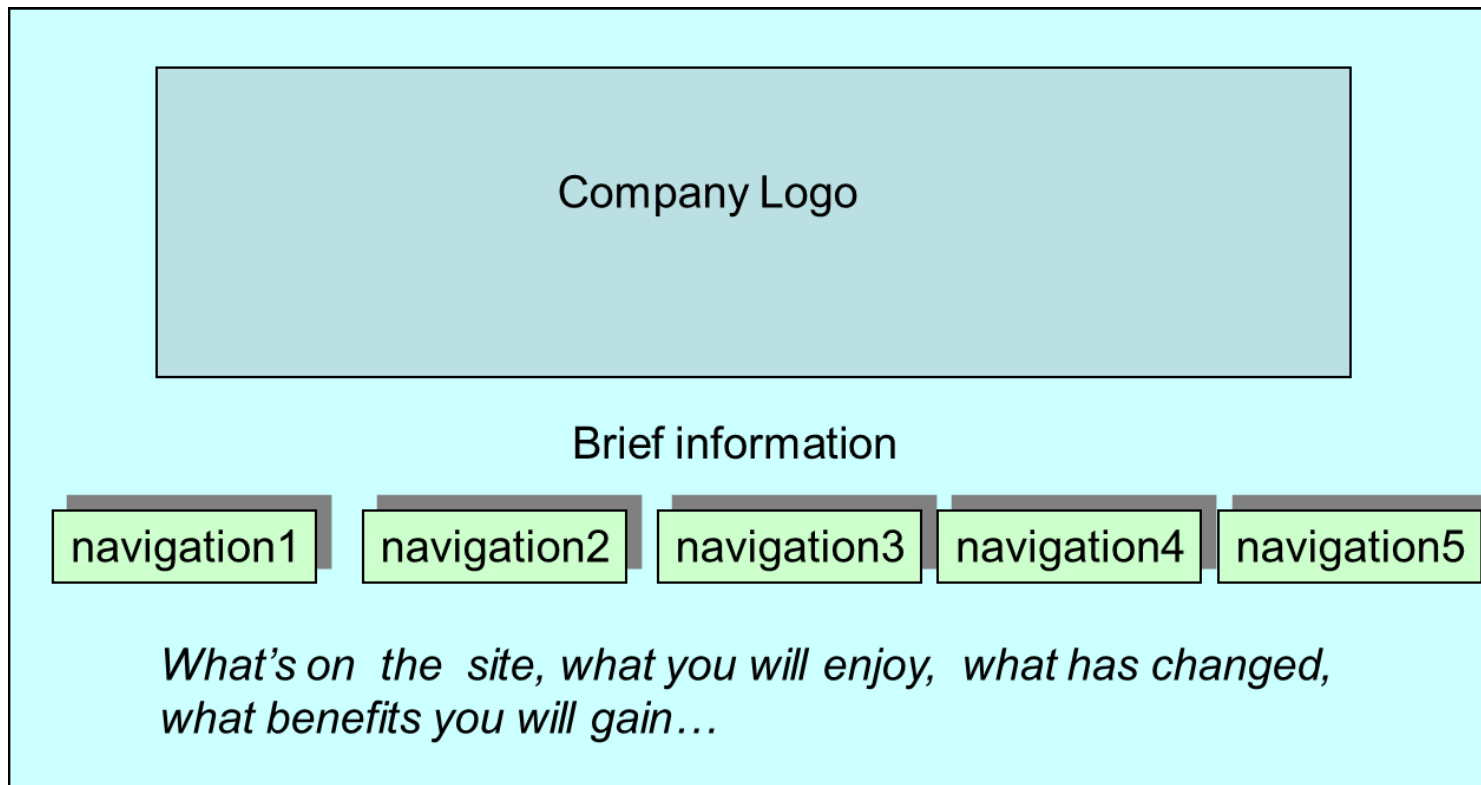
- Breadcrumbs tell the user where they are relative to the home page
- They are secondary to navigation bar
- They may include a search box

The popular 3-region Web page layout

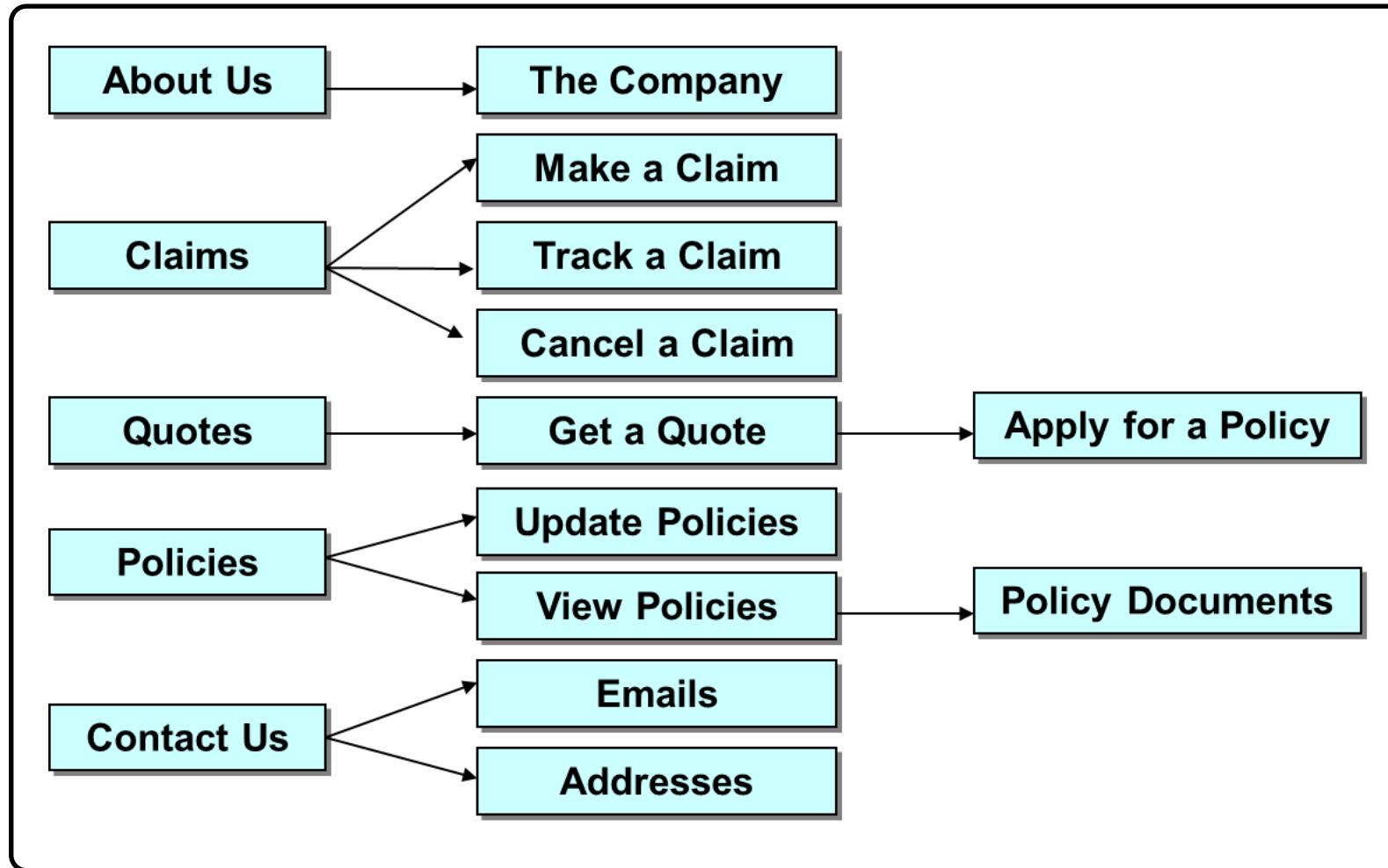


The Home Page

- Can be an exception to the three-region layout
- But not a splash screen – needs to include navigation to the main use cases



Site Map - Example from Case Study Project



Store Content in the Database

- Some content is used in many different places and/or can change frequently



General Website Design Guidelines

- Design around existing content, not future content
- Avoid unnecessary images
- Exploit hyperlinks
- Use Cascading Style Sheets
- Make navigation flow
- Visit your own site regularly

(End:Section 4)

End of Topic

- Feel free to:
 - *ask any Questions via Stream Forums*
 - *also send questions to the course instructor but you are encouraged to use forums so that answers could benefit the rest of the class*

Tutorial I - Topic Review Case Study

- A Web Site Development Project
- Project title: **Online Home Insurance**
- An example of *applying analysis and design* to Website development

Project Scenario

Web Home Cover is a new enterprise set up to provide home insurance over the web. The business case is based on providing a service that is entirely on-line and therefore highly efficient in terms of the initial capital investment required by the insurance company. Since the company will only operate via the web, it must have a web application that meets the needs of all its customers and staff. It must also be written to ensure that it will work for as many web clients as possible, from desktop computers to mobile devices

Mission Statement

To bring home insurance services to every corner of the Web

Requirements Gathering

- Focus group
- User profiles / personas
- Joint requirements workshop (of stakeholders)
- Brain storming / card storming
- 12 or so core requirements

Prioritizing Requirements

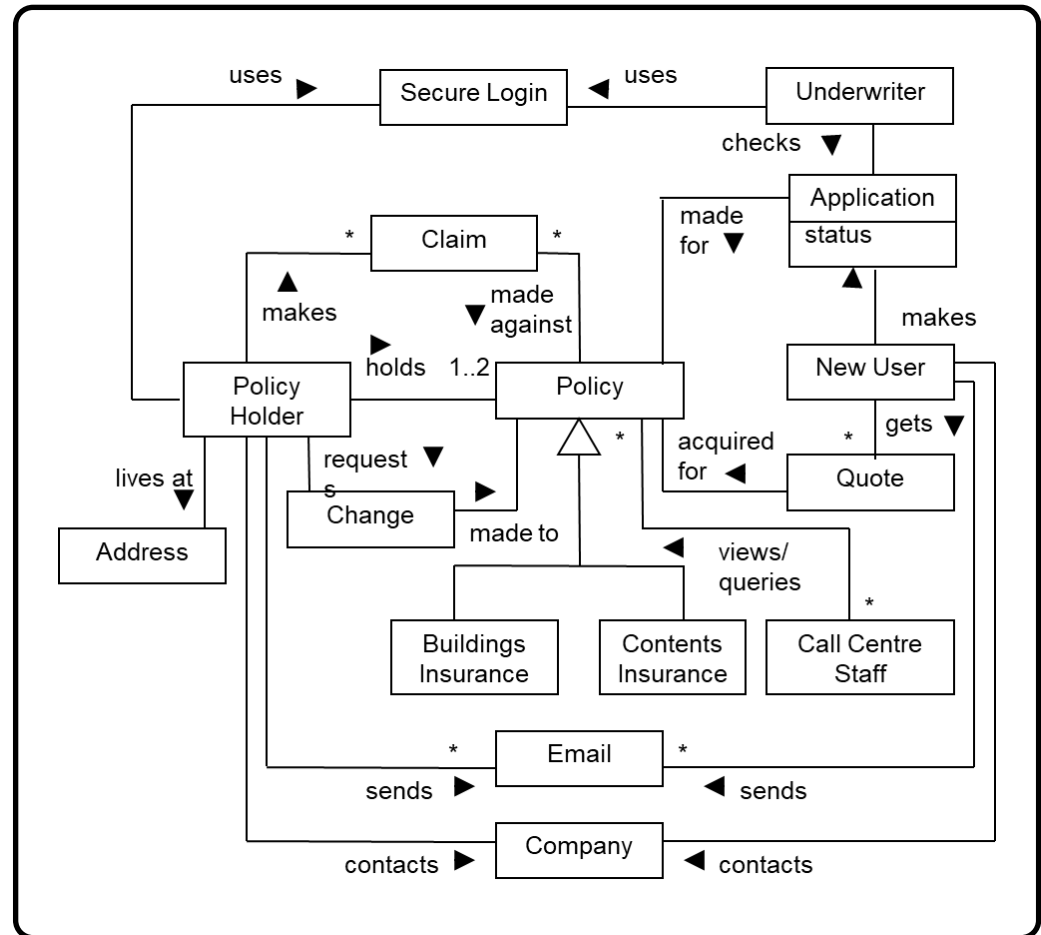
- **MoSCoW:**
 - *Must have*
 - *Should have*
 - *Could have*
 - *Want to have*
- Can decide by voting, using multiple votes from different perspectives

Concept List

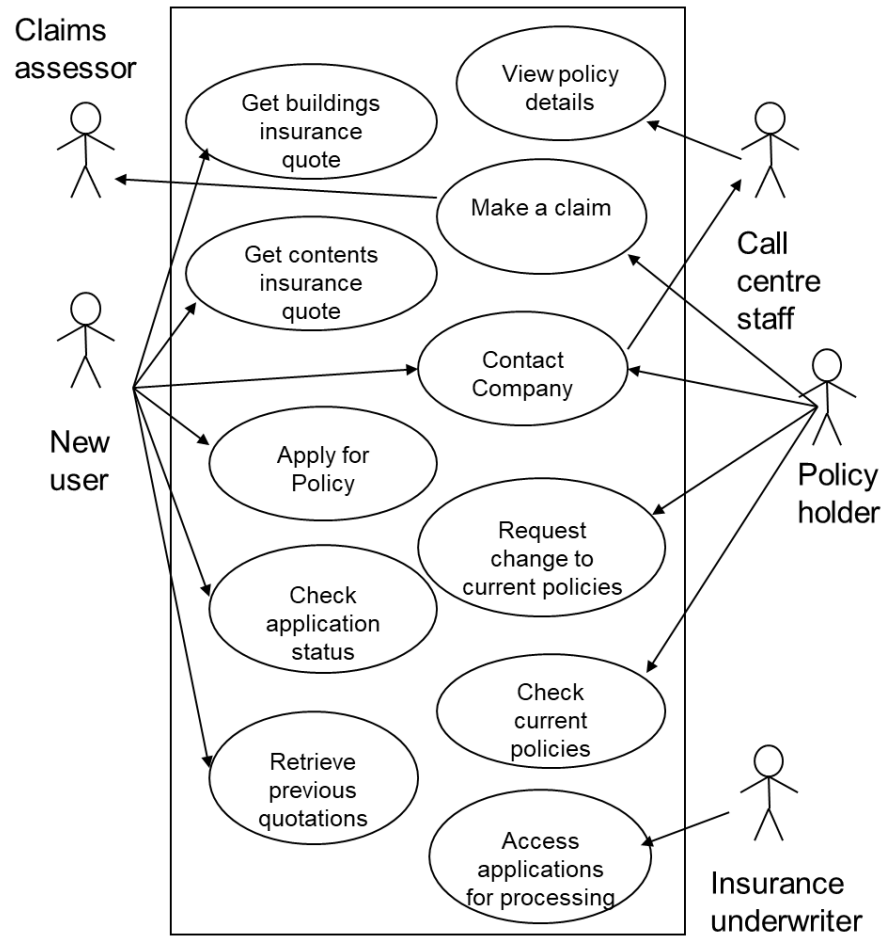
- Identified from the core requirements
- Remove concepts from the list that are:
 - *Outside the system boundary*
 - *The boundary itself*
 - *Nouns for the system as a whole*
 - *Synonyms*
 - *Properties of other concepts*

Domain Model

- Describes the following:
 - *Key concepts from the domain*
 - *Which components interact with each other*
 - *How these relations can be described*
 - *The cardinality of interaction:*
 - one to one,
 - one to many, and
 - many to many.



Use Case Diagram



Use Case Description

Use Case Name : Get Buildings Insurance Quote

Actors: New user

Start page: Home page

Use Case Description:

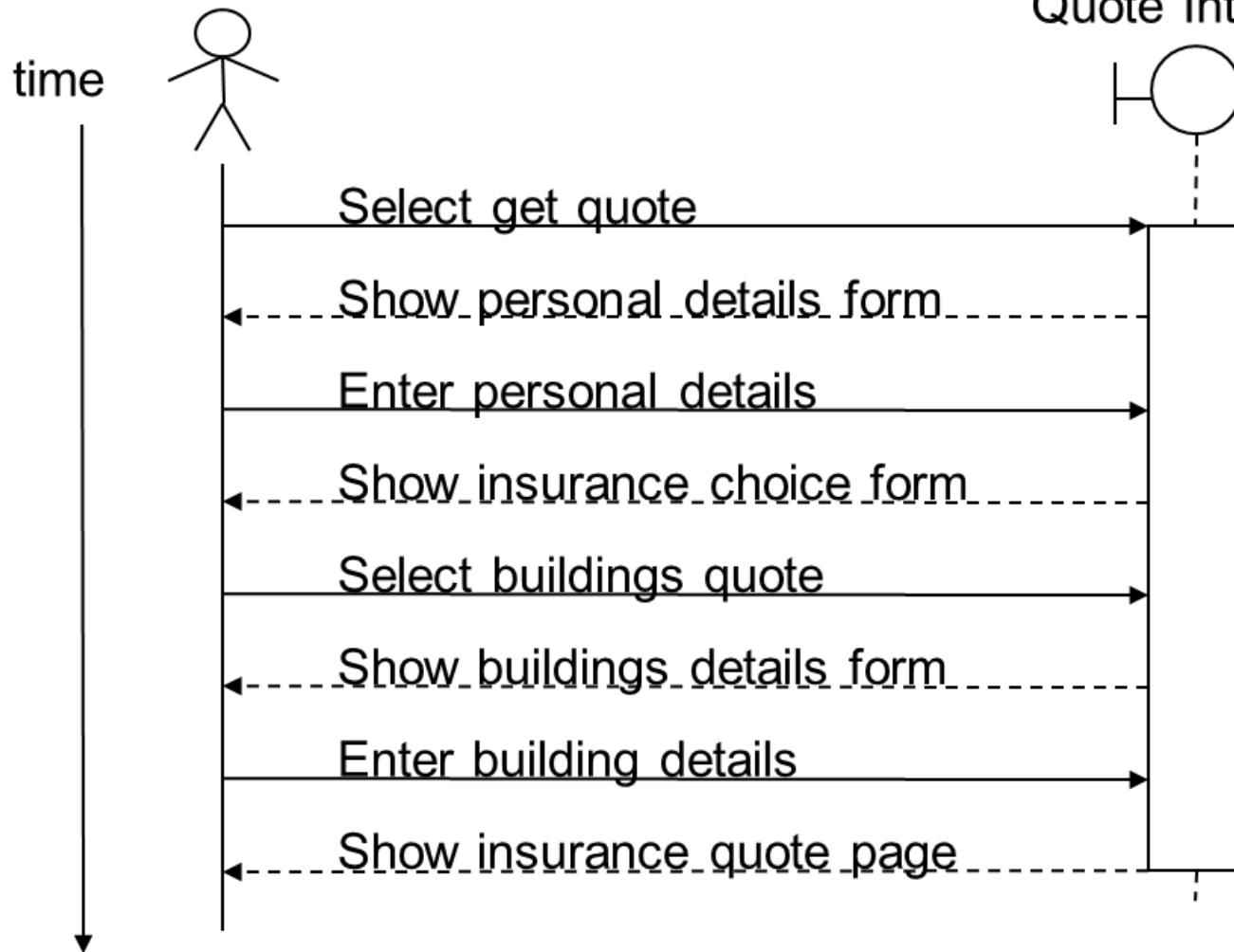
1. The actor chooses to get an insurance quote
2. The system requests the actor's personal details
3. The actor enters his/her personal details
4. The system displays a choice of available insurance quotes
5. The actor chooses to get a buildings insurance quote
6. The system requests information about the building to be insured
7. The actor enters data about the building
8. The system displays the buildings insurance quote.

System Sequence Diagram

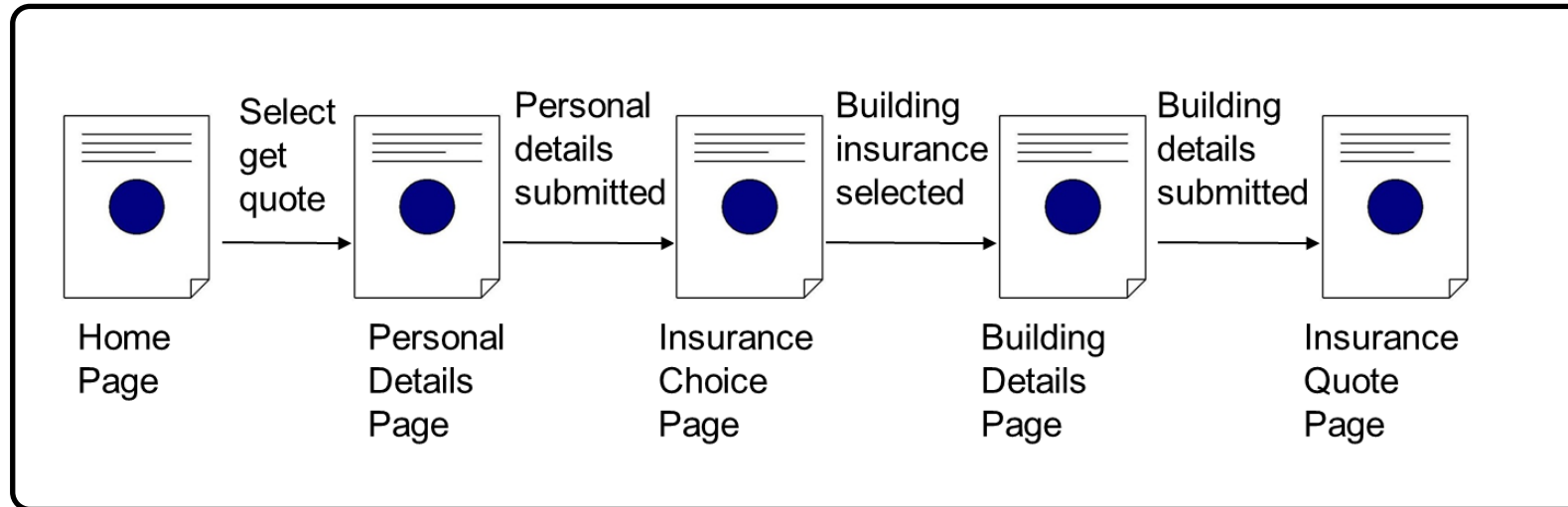
New User

Insurance
Quote Interface

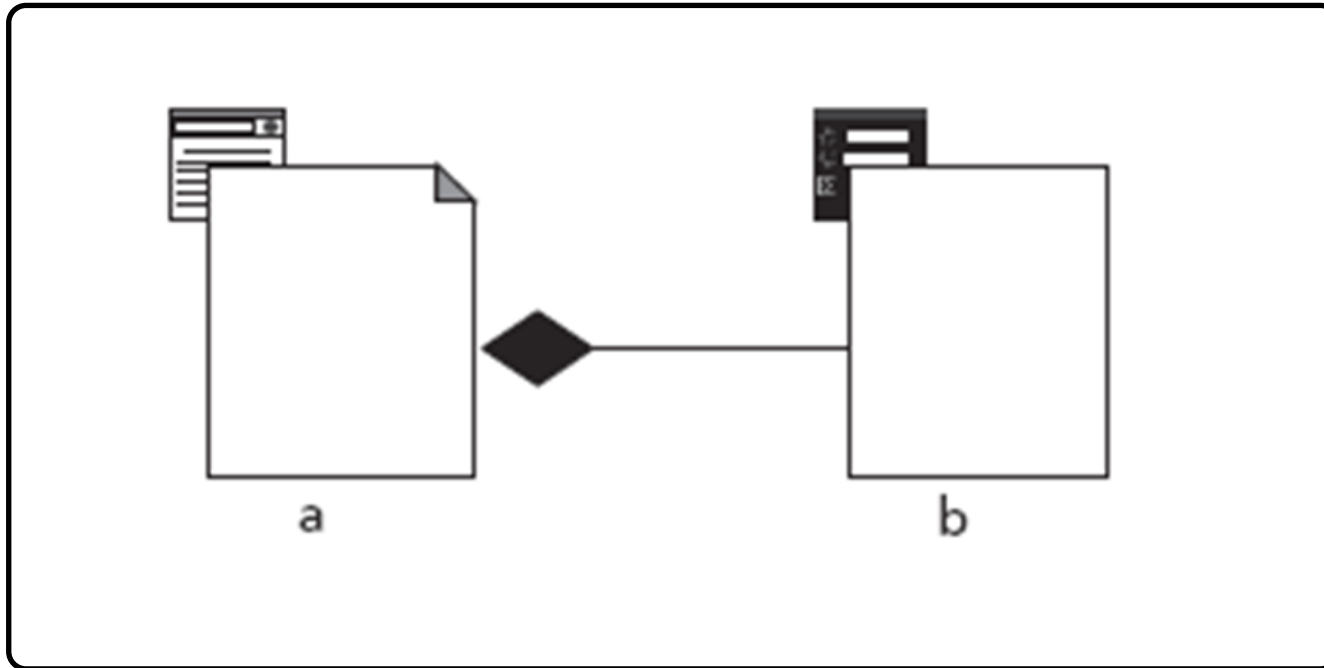
time



Basic Story Board

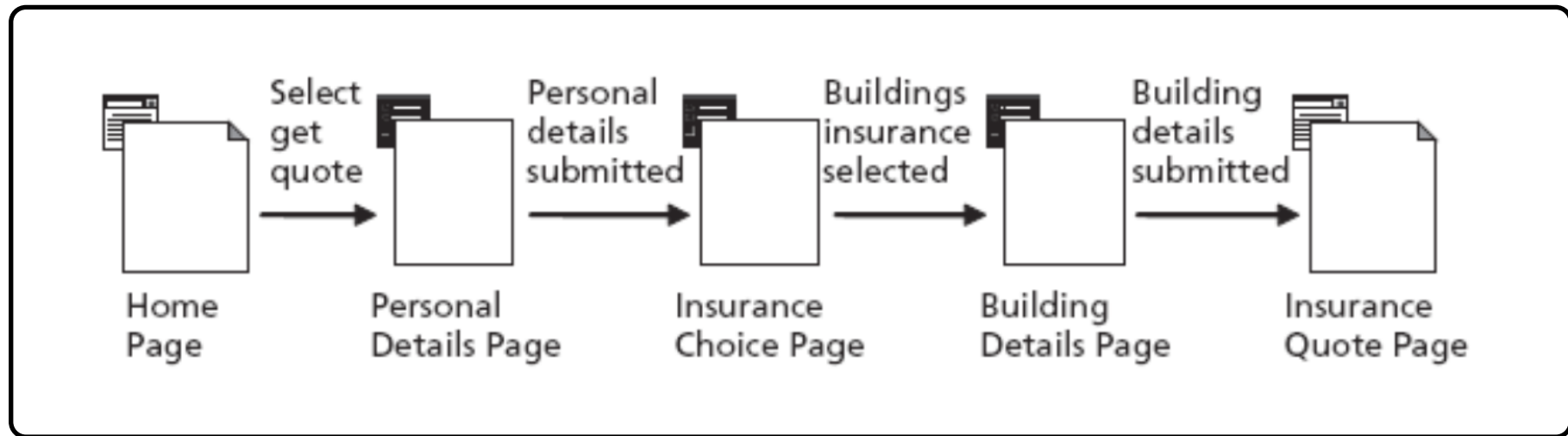


UML Extensions for Web App Design



*Symbols for a client page (a) that contains (shaded diamond)
a Web form (b)*

Modified Storyboard



This version of the storyboard uses specific page types

Alternate Flow

Alternate flow – contents insurance only

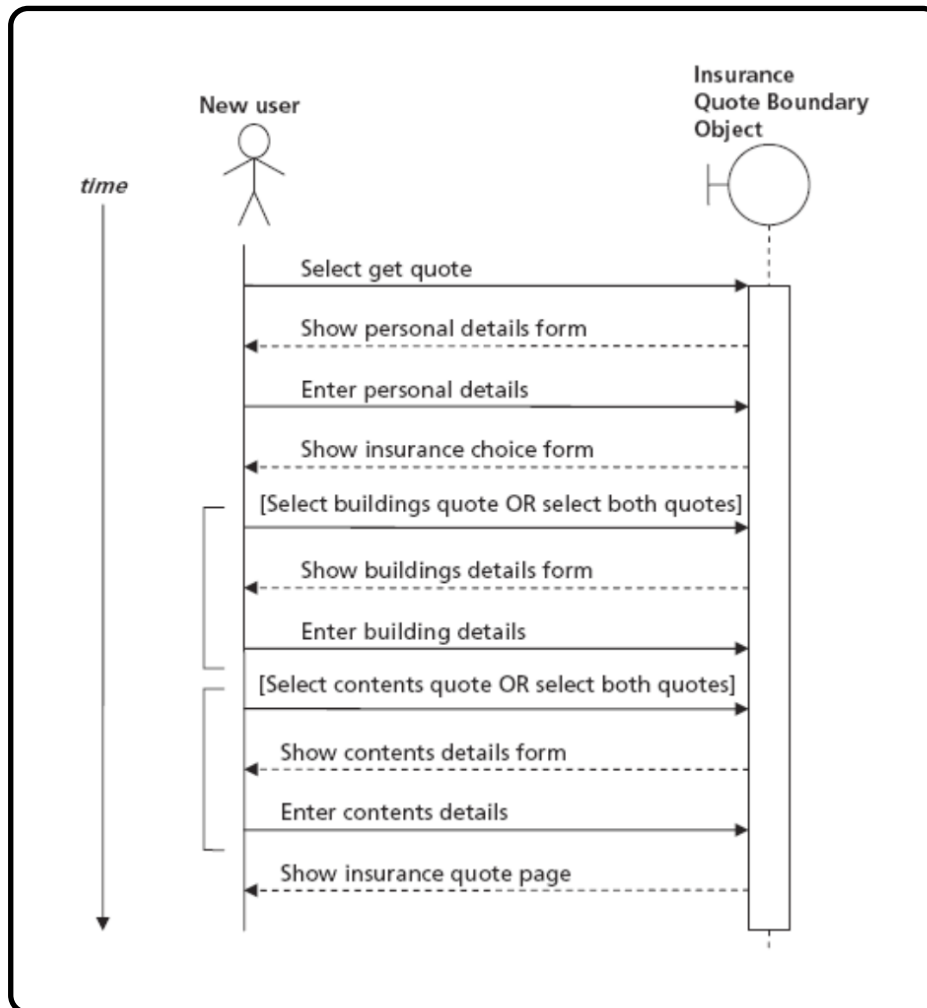
- 5a. The actor chooses to get a contents insurance quote.
- 6a. The system requests information about the contents to be insured.
- 7a. The actor enters data about the contents.
- 8a. The system displays the contents insurance quote.

Alternate flow – both types of insurance

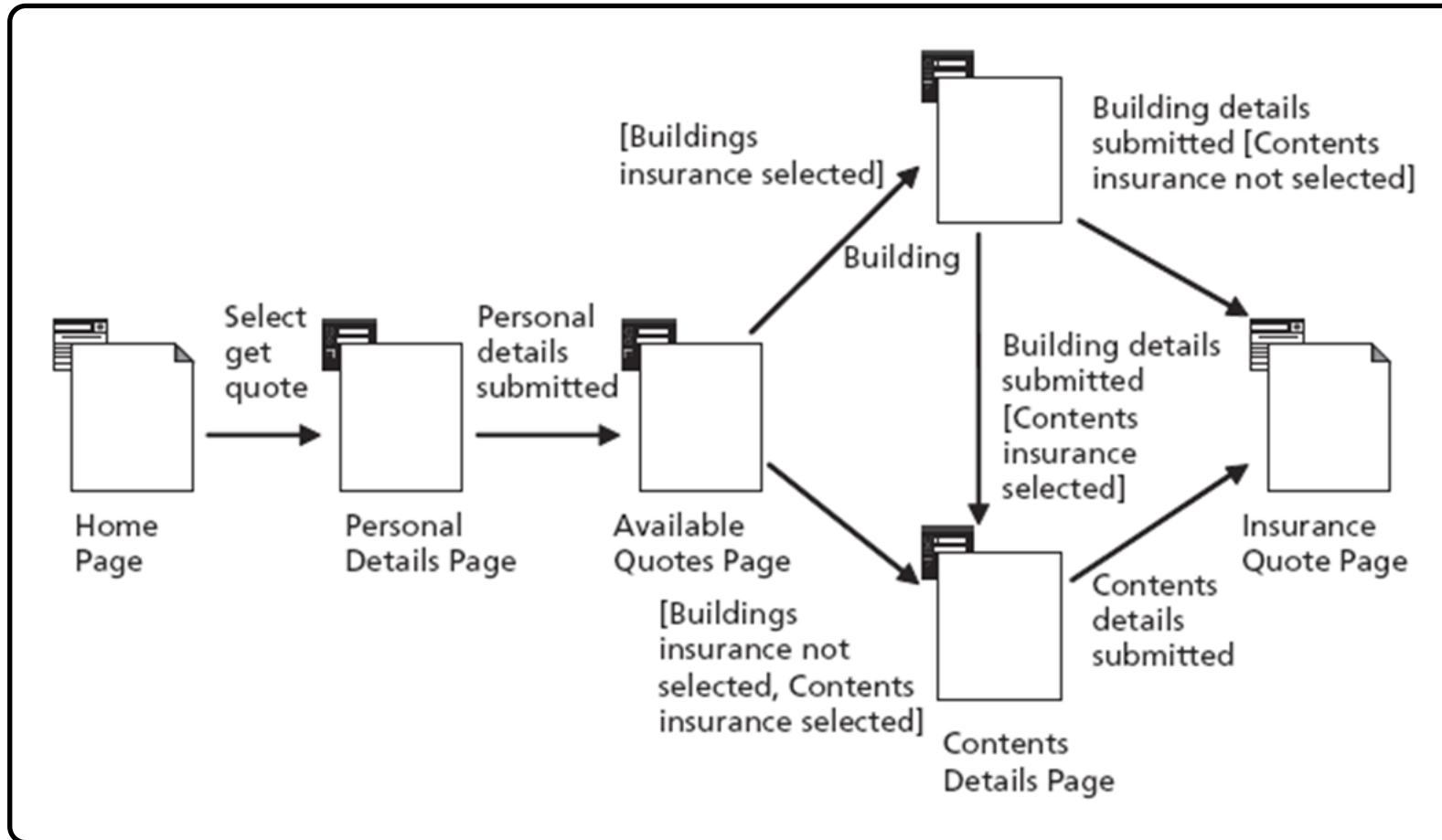
- 5a. The actor chooses to get both a buildings insurance quote and a contents insurance quote.
- 6, 7, 6a, 7a
- 8b. The system displays a contents insurance quote, a buildings insurance quote and a total.

This description is added to our existing use case description

Modified Sequence Diagram



Updated Storyboard



Tutorial Challenge

1. Search the Web for “website storyboard diagram tool” and create a list of 5 tools with Web links for drawing website storyboards. Explore each and select your favourite. Post a link to your favourite tool on 158.258 course channel and invite comments or any feedback from the class.
 2. Using the tool you selected in (1) above, draw a storyboard diagram for a 13-year student at a high school who would like to apply to Massey University to study for the Bachelor of Information Science degree. Save your diagram as png with your student ID as filename. Post your storyboard for feedback to the course discussion forum on Stream.
-