



답변형 게시판

zelratole@hanmail.net



Contents :

1

Click to add Title

2

Click to add Title

3

Click to add Title

4

Click to add Title



스프링 개발환경처리 :



주요 기능 :

- 초기 글 리스트 화면
 - 답변형 계층구조
 - 페이지 처리
- 등록화면
 - 기본 글 등록 처리
 - 첨부파일
- 상세화면
 - 수정, 삭제
 - 답글 달기



구현 순서 :

- 화면 ui 결정 및 프로세스 결정
 - 초기 화면(html,css)
- DB 설계
 - ERD, table 생성
- 각 화면단에 활용할 sql구현
 - 계층형 sql 구현(답변처리)
- 위 sql에 매핑되는 VO, DTO 클래스 구현



확인예제(문제는행게시판):

- A02_QuebankCtrl.java
- a11_quebankList.jsp
- table QUEBANK 활용



초기 화면 구현 :

- ctrl
 - 공통 RequestMapping를 클래스명 위에 선언
 - 각 기능별 메서드에 `params="method=list"`,
- view단
 - 검색항목
 - 제목, 작성자, 내용
 - list항목
 - 번호, 제목, 작성자, 작성일



- 관리할 데이터 list
 - 리스트 테이블: num(키), 제목, 내용, 비밀번호, 작성자, 읽은 수, 등록일, 이메일
 - 첨부파일 테이블 : num(키), 파일명, 생성일
- sequence : board_seq



게시물 등록 :

- 프로세스
 - 메인 화면에서 등록 버튼 클릭
 - 등록 화면
 - 제목, 내용, 비밀번호, 작성자, 이메일
 - 등록 처리
 - 메인 화면에서 등록된 list 확인..
- 개발 순서
 - 초기 웹 화면구성
 - DB sql 작성..
 - DTO 구성 내지 선택
 - DAO(Repository)
 - 화면 구성 : dto(property)연결되는 name 부분
 - Controller 연결 처리..service, dao



초기 화면 구성 및 등록 :

- main에서 연결하는 등록화면 Controller 으로 연계되어 처리할 수 있도록
- sql 구현

```
insert into board
values(board_seq.nextval,0,'게시판등록',
'냉무', '1111', '홍길동',0,sysdate,'hihi@hanmail.net' )
```
- DTO 구성 내지 선택
 - Board.java
- DAO(Repository)
 - 입력:Board return : void
 - package 등록 여부 확인
 - dispatcher-servlet.xml에 등록 여부 확인(,로 패키지 등록)
- mybatis : config, mapper



스프링 개발환경처리 :

- 개발환경구성방식
 - maven 자동처리
 - lib 추가 처리
- 프레임워크구조
 - web.xml
 - spring프레임워크를 사용할 수 있게 끔. 호출 처리
 - config : 스프링 프레임워크의 설정 위치
 - 한글처리 filter
 - dispatcher.xml
 - 스프링의 환경 설정 구체적인 내용
 - mapping(model, view, controller), lib, database등 처리
 - 필요한 lib의 설정 위치를 지정 - mybatis
 - resource에서 상세 처리 지정



- dispatcher.xml에서 controller 등록
- controller
 - 스프링의 공통 servlet인 DispatcherServlet의 다음 단계의 내용을 처리..
 - 요청된 내용 받고, model단으로 요청처리에 필요한 business logic을 처리
 - Service, Dao
 - view단에 넘겨줄 model를 매핑시킨 후,
 - view단 호출
- View단 jsp호출



스프링프레임워크 프로세서 :

- 시작
 - <http://localhost:8080/springweb2/start.do>
- WEB-INF 하위에 web.xml 호출
 - 공통 스프링 모듈 dispatcher Servlet
 - **servlet name**으로 등록된 이름을 기준으로
 - **servlet이름.xml**을 WEB-INF에서 설정파일로 찾음
- **dispatcher-servlet.xml** (스프링관련설정)
 - 웹 컨테이너 로딩 시, **controller**를 메모리에 등록
 - 필요한 모듈 선언해서 메모리에 **loading**
 - **viewresolver**: 화면단 호출에 대한 방식 처리
- **mapping handler**를 통해 **start.do**를 **controller** 내에서 찾음.
 - @RequestMapping("start.do")



Controller:

■ 구조

- `http://localhost:8080/springweb/mapping이름`
- `@Controller` 클래스명 위에 선언
- `@RequestMapping("mapping이름")`
 - 컨트롤러에 바로 선언, 메서드에 선언
- 메서드 선언규칙
 - `public String` 메서드명(외부전달값, Model d)
 - 모델단 처리
 - `return View`단 호출
 - ex) `start.do?id=홍길동&pass=7777`



controller단 입력과 호출처리 :

- /springweb2/start.do?id=홍길동&pass=7777
- @RequestMapping("호출하는 mapping이름")
- public String form(입력값 형식)
- 입력값 형식
 - @RequestParam("key값") 할당받는 데이터 형식 변수명
 - ex) @RequestParam("id") String name
 - @RequestParam("pass") int pass
 - VO :Member에 setXXX,getXXX 있으면 자동으로 객체에 입력처리 됨..
 - Member mem
 - mem.getId(), mem.getPass() 데이터가 입력되어 있음.
 - ex) public String form(Member mem){
 -
 - }



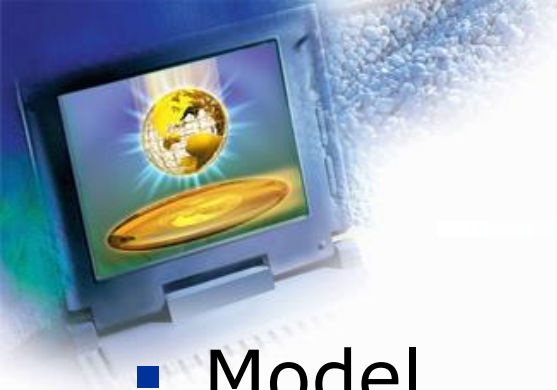
확인예제 :

- controller
- A03_ParamExp.java
- view a03_ParamExp.jsp
- url
http://localhost:8080/springweb2/param2.do
- * ?procname=사과&price=3000&cnt=3
- 출력물 물건명 :@@@ 가격 :@@@ 갯수 :@@@ 총계 :@@@



VO값으로 전달될 때 :

- 요청 url
 - `param2.do?id=홍길동&pass=7777`
- Controller단에서 처리 메서드
 - `@RequestMapping("param2.do")`
 - `public String form(Member mem){`
 - `:처리 내용 mem.setId("홍길동")`
 - `mem.setPass(7777)`



View단으로 데이터 넘기기 :

- Model


- 사용위치

- `public String form(Vo v, Model d)`

- 메서드

- `d.addAttribute("key", 넘겨줄 obj);`

- View 단 `${key.메서드}`

- 
- 요청 VO값 및 view 전달처리
 - 두개의 값을 전달 받아서
 - url : param04.do?num01=25&num02=27
 - VO로 입력
 - controller 단에서 연산 처리
 - View단에 + 의 결과값을 전달
 - 화면단
 - $25 + 27 = 52$



- spring에서 mapping처리..



method 방식 확인예제 :

- Controller:
 - A02_MethodExp.java
- View:
 - a02_methodexp.jsp
- 처리 내용
 - 초기화면 : ID:[] [로그인]
 - 요청후화면:
 - controller단에서 id값 himan 인증 로직처리
 - @@@님 환영합니다/인증된 계정이 아닙니다.

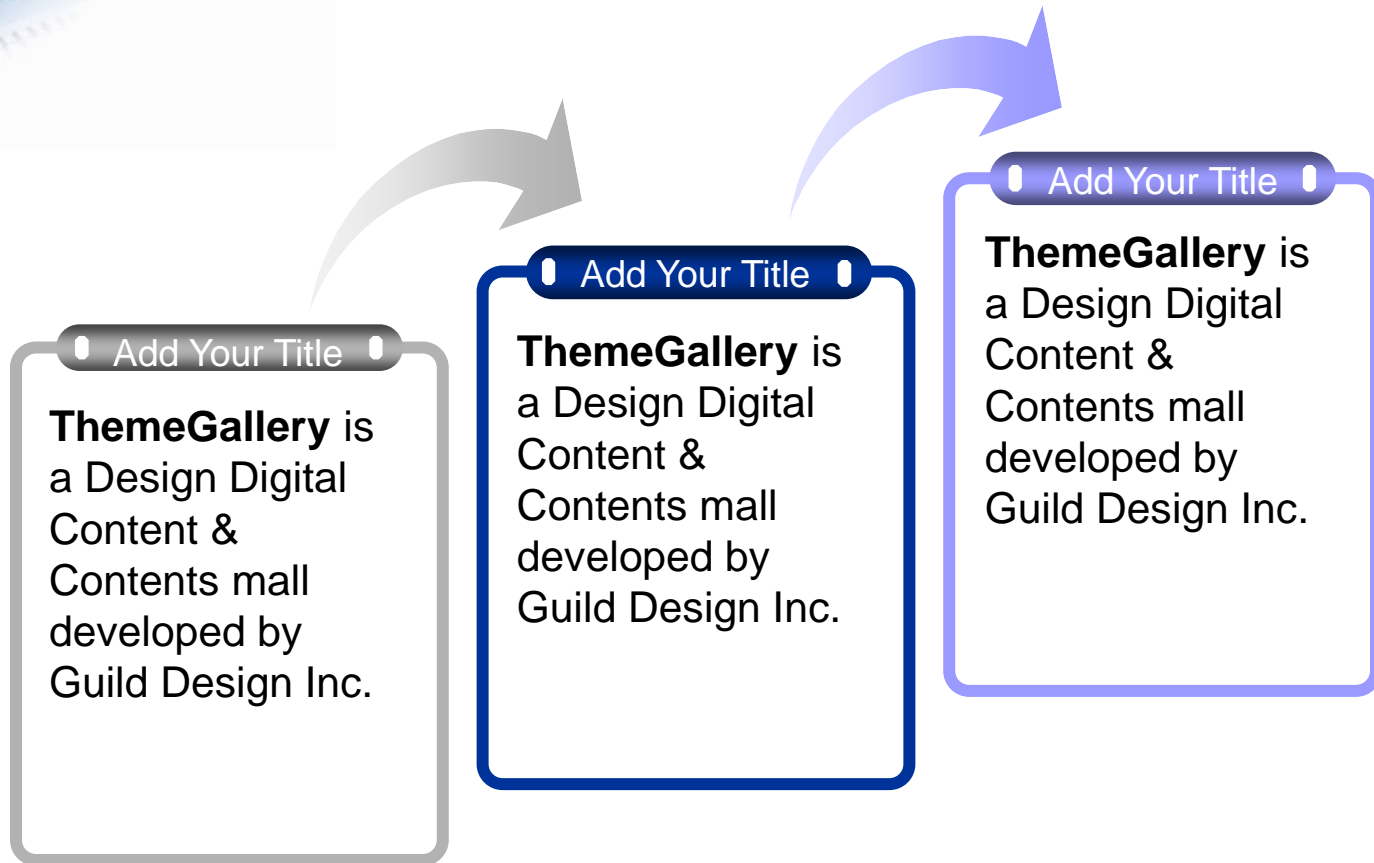


요청내용 메서드별 분리 :

- 스프링에서 Controller의 생성기준 한 메인화면에 같은 데이터를 보면서 업무처리(CRUD)가 일어날 때는 기준으로 한다.
- `http://...../springweb2/board.do?@@@@`
- `@RequestMapping("board.do")`를 @controller과 함께 클래스명 위에서 선언.
- 각 메서드 별로
 - `@RequestMapping(params="method=list")`
 - `@RequestMapping(params="method=insert")`
 - `@RequestMapping(params="method=update")`
 - ex) `springweb2/board.do?method=list`
- View단
 - 각 프로세스 버튼 이벤트가 호출될 때..
 - `$("#insBtn").onclick(function(){`
 - `$("form").attr("action","${path}/board.do?method=insert");`
 - `});`



Diagram :





Marketing Diagram :

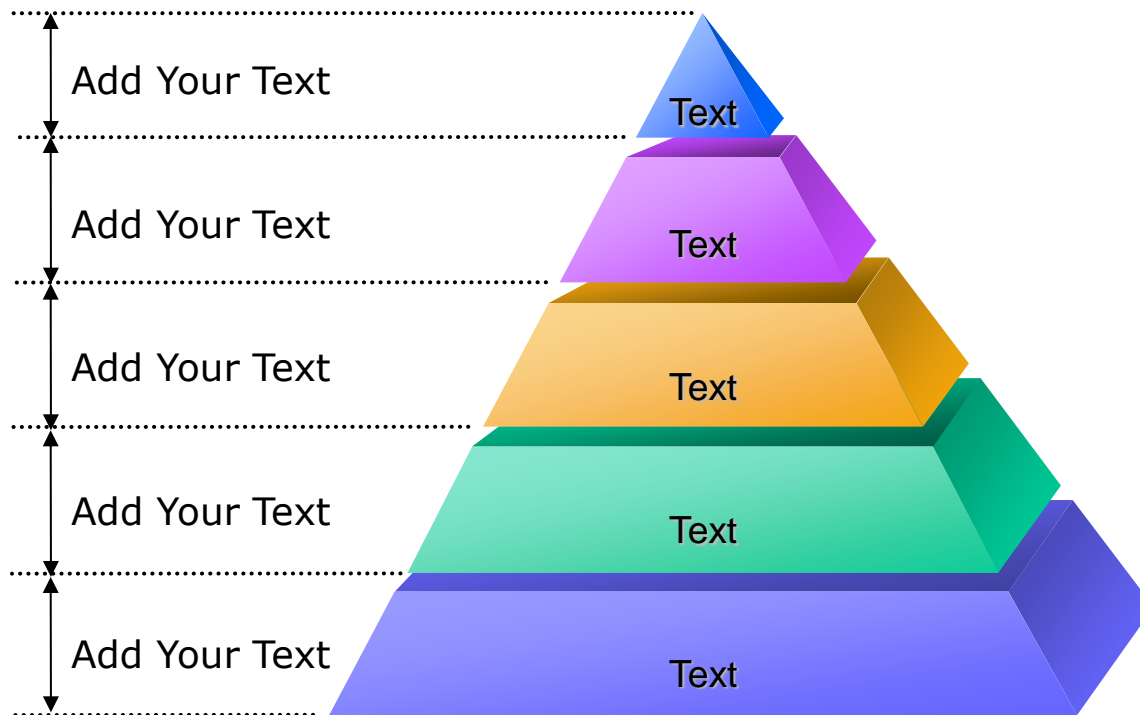
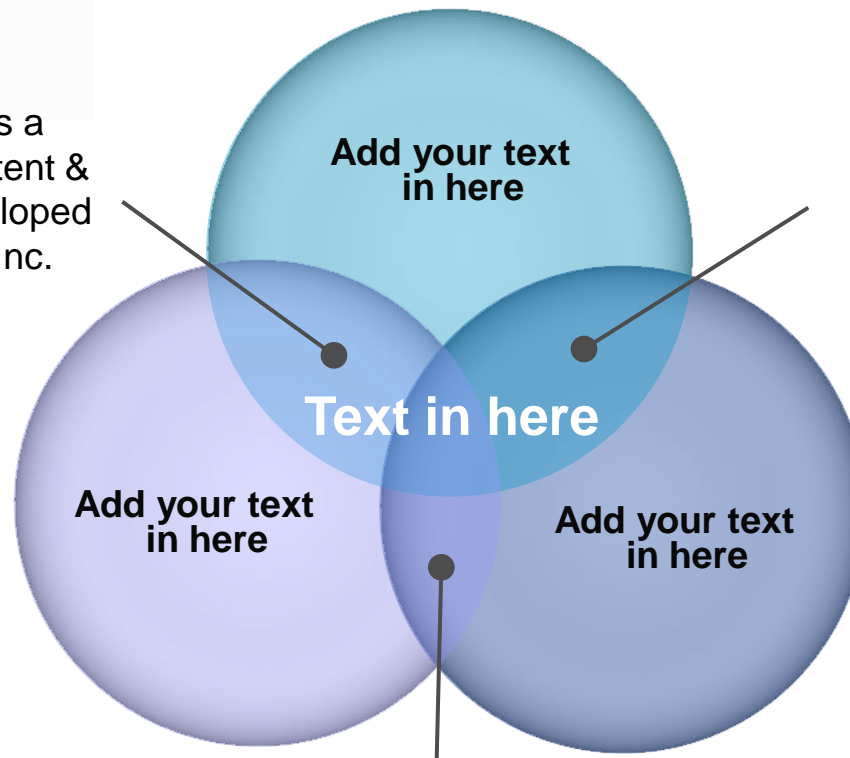




Diagram :

ThemeGallery is a
Design Digital Content &
Contents mall developed
by Guild Design Inc.



ThemeGallery is a
Design Digital Content &
Contents mall developed
by Guild Design Inc.

ThemeGallery is a
Design Digital Content &
Contents mall developed
by Guild Design Inc.



Diagram :

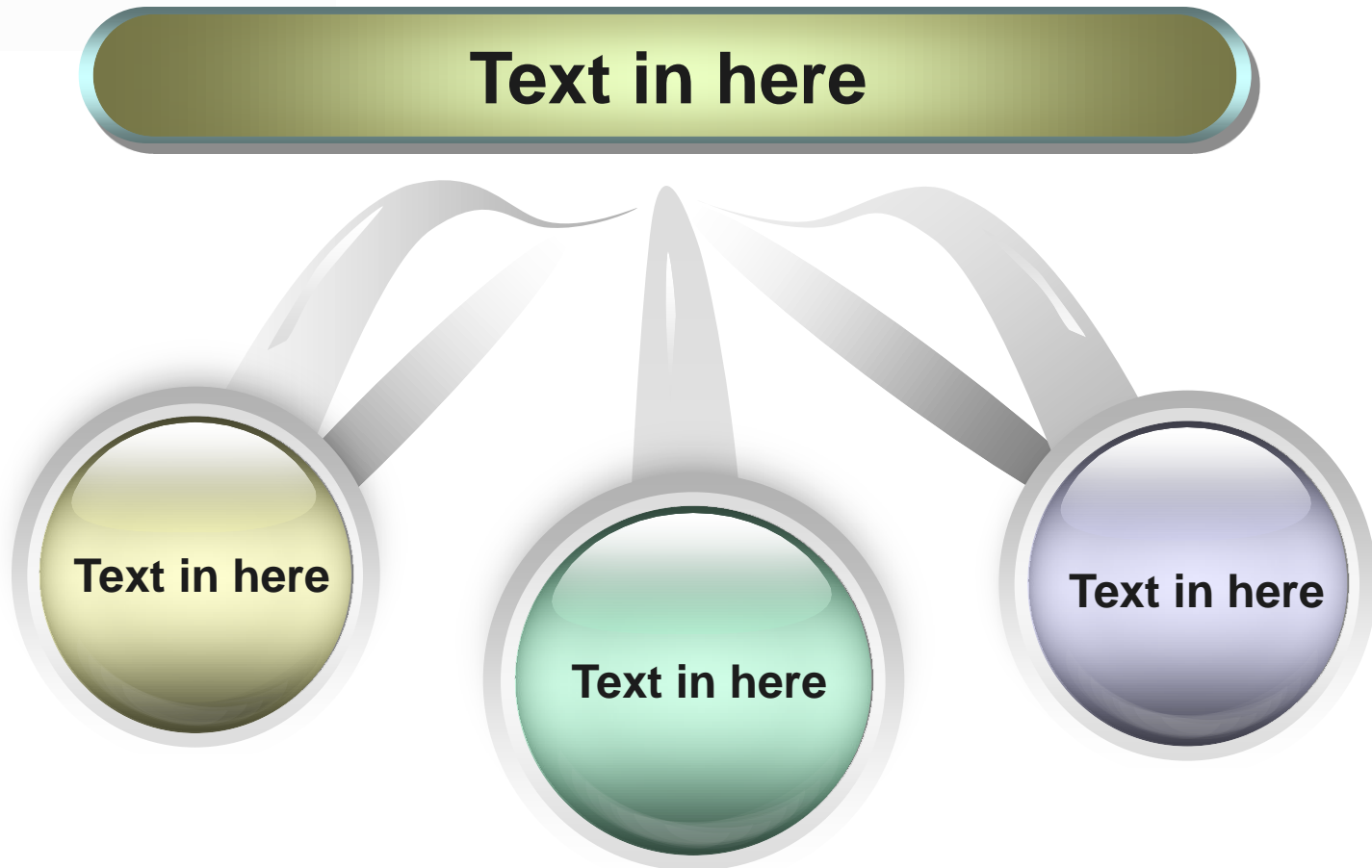




Diagram :

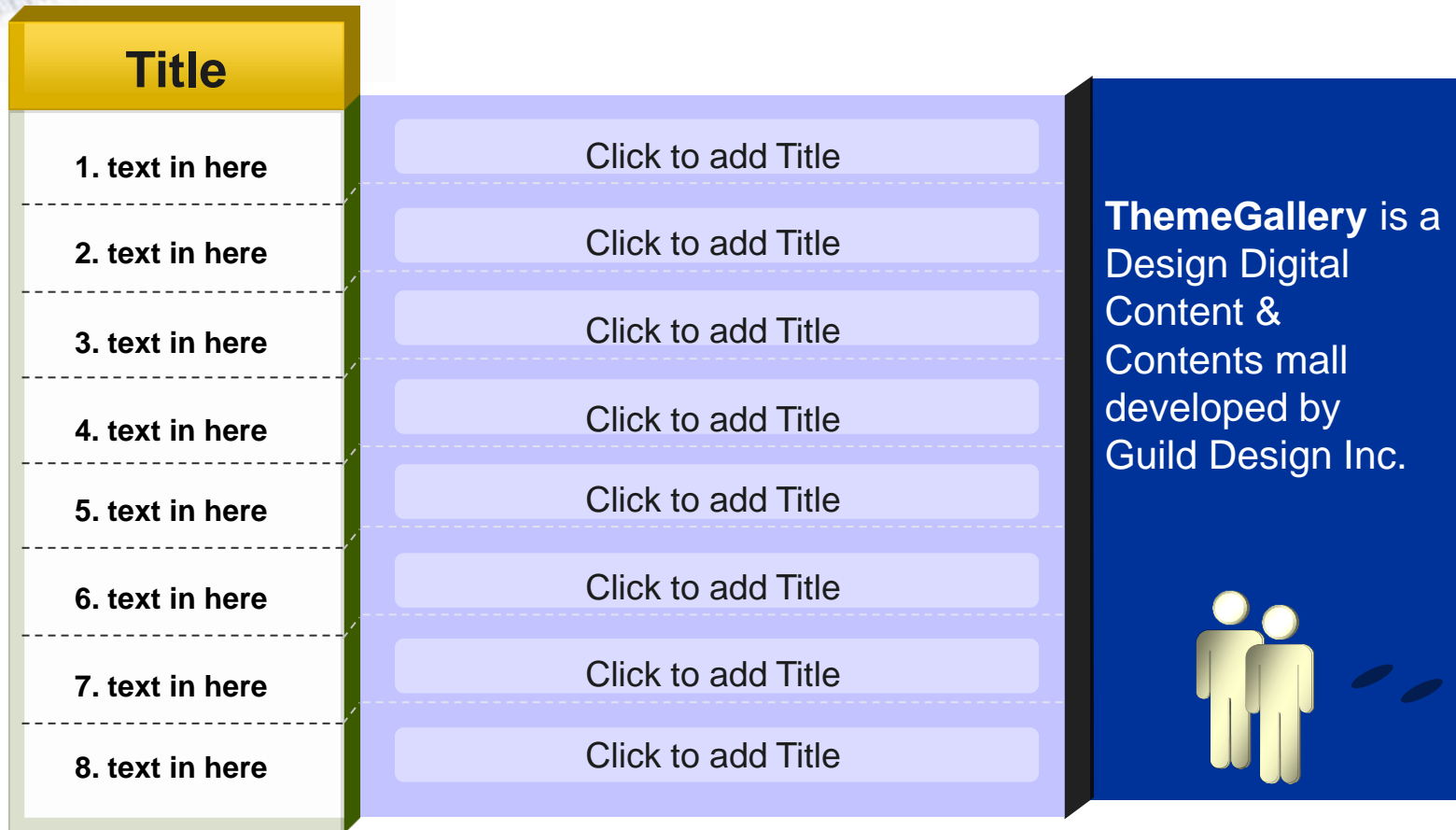
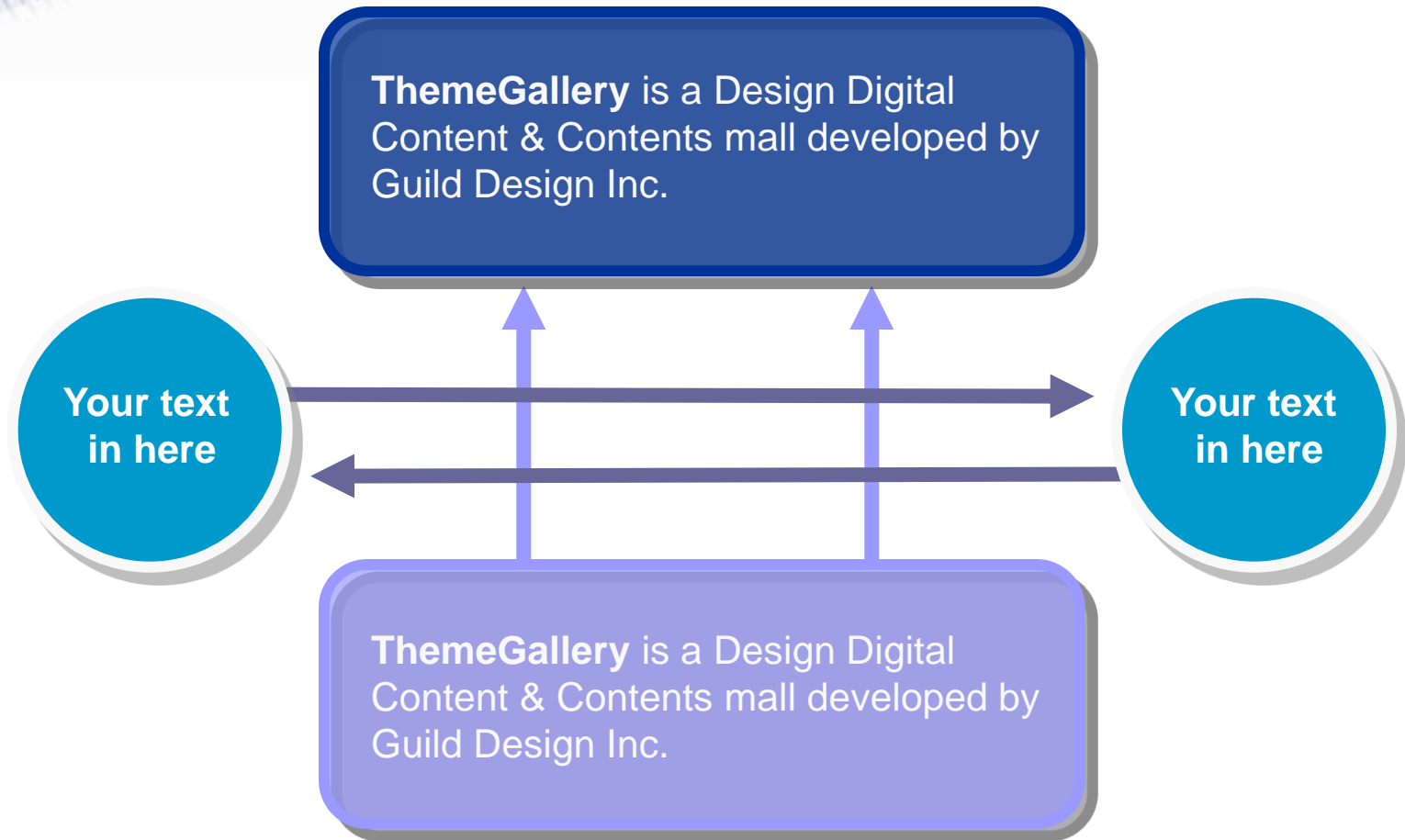




Diagram :





Thank You !

zelratole@hanmail.net