

## Design Plan Exercise

### Goal

Create a simple design for your social networking application. Collaborate with others but do not coast on their effort. You must provide as many ideas as you consume.

Use the World Press Design plan as a model for your plan for the Social Networking app.

Define the follow levels:

- Architecture
- Interfaces
- Data

Submit your results

---

### Architecture

Architecture - Top-level system design

- Interfaces
- Building blocks

Block Diagram

- Mobile clients
- Computer clients
- Web server
- Application server
- Database

Architecture Process

- Partition the system
- Define interfaces
- Create a test plan
- Manage dependencies

Interfaces

- Each interface can be described as a stack of functions
- Define the signatures

### World Press System

Architecture

- Computer client - Web browser
- Web server - http, urls to web pages
- Email server - SMTP
- Application server - function interface
- Database - CRUD for User, Article, Subscriber

Web Server Interface

- Login
  - register (name, email, password)
  - login (name, password)
- Author
  - add\_post (user, title, body)
  - edit\_post (user, title, body)
  - delete\_post (user, title)
- Reader
  - get\_post (title)
  - list\_posts (user)
- News
  - register (email)
  - verify\_email (token)
  - unsubscribe (email)

Application Server Interface

- User
  - CRUD - create, read, update, delete
- Article
  - CRUD - create, read, update, delete
- Subscriber
  - CRUD - create, read, update, delete

Database Schema

- User
  - name, email, password
- Article
  - user, title, body
- Subscriber
  - email, user

---

### Interfaces - Detailed Design

Start with Architecture (High-level Design)

- Subsystems - partitioning of components
- Interfaces - every subsystems has at least one interface
- Data and control flow - walk the call chain

### World Press Web Server Interface

- Login
  - register (name, email, password)
  - login (name, password)
- Author
  - add\_post (user, title, body)
  - edit\_post (user, title, body)
  - delete\_post (user, title)
- Reader
  - get\_post (title)
  - list\_posts (user)
- *News*
  - *register (email)*
  - *verify\_email (token)*
  - *unsubscribe (email)*

### World Press Application Server Interface (Functions)

- User
  - create\_user (name, email, password)
  - read\_user (name, password) --> user\_id
  - list\_user () --> user list
  - update\_user (name, email, password)
  - delete\_user (user\_id)
- Article
  - create\_article (user\_id, title, body)
  - read\_article (article\_id)
  - list\_article (user\_id)
  - update\_article (user\_id, title, body)
  - delete\_article (user\_id)

---

### Data

ORM

- Define your database structures by specifying classes.
- Automatically generate SQL to match the objects.
- All work is done in the object domain.

### World Press Data

- User
  - name, email, password
- Post
  - user, title, body
- Subscriber
  - email, user

### World Press Classes

- User
  - name, email, password
- Post
  - user, title, body
- Subscriber
  - email, user

### World Press Functions for Data

# Article

```
def create_article (user_id, title, body):  
    Article.object.create (user_id, title, body)
```

```
def list_article (user_id):  
    return Article.objects.all()
```

```
def read_article (article_id):  
    return Article.objects.get(pk=article_id)
```

```
def update_article (user_id, title, body):  
    a = Article.objects.get (user_id=user_id, title=title)  
    a.body = body  
    a.save()
```

```
def delete_article (user_id, title):  
    Article.objects.delete(user_id=user_id, title=title)
```