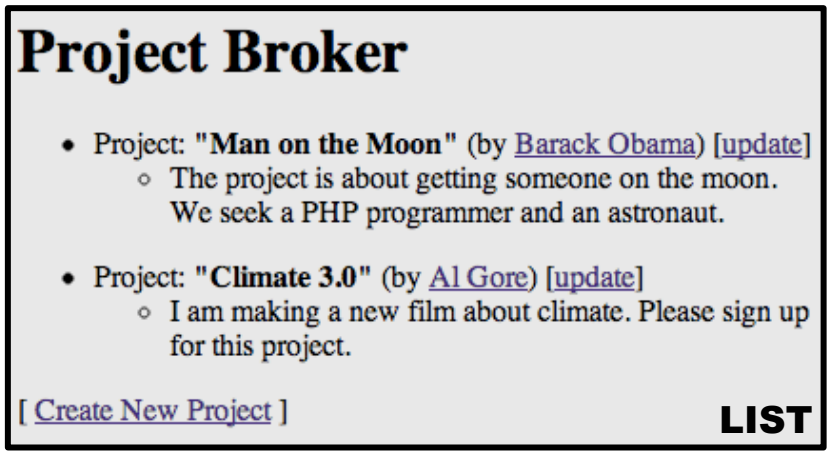
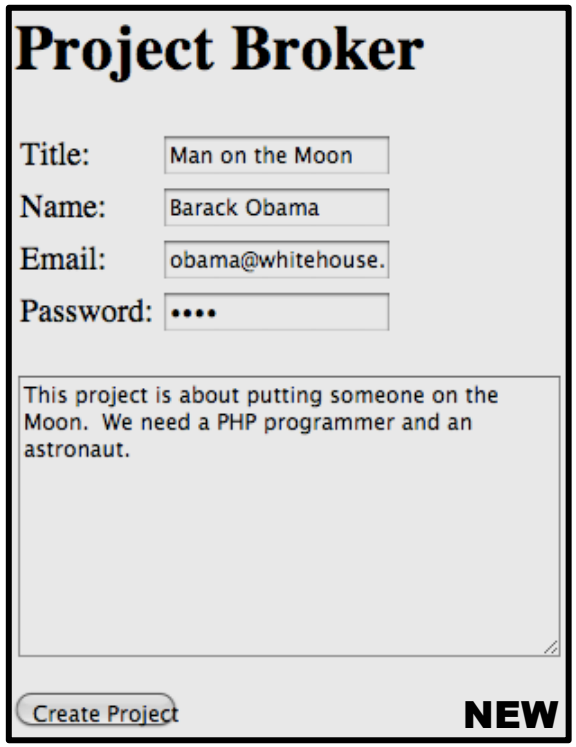
A9: Project Broker

4-Step design process:

1) Design data model
2) Develop data transactions
3) Develop site map and forms
4) Program it all in SQL+PHP

Fairly Open Assignment





A9.1: Design Data Model

Destination file: **`A9/project_table.sql`**

Define an SQL table using the CREATE TABLE statement which contains all the necessary fields, guided by the following description:

The table should contain information about the title of the project, a short project description, and a project editing password. It should furthermore contain info about the project administrator (name, email). Furthermore, each project should have a unique numerical id (use PRIMARY KEY and AUTO_INCREMENT). Finally, each project title should be forced to be unique by declaring that field as UNIQUE.

Add two projects to your table using suitable INSERT INTO commands.
Save your CREATE TABLE and INSERT INTO statements in the file
`A9/project_table.sql`

*(Note: It would probably be better to split the table up into two tables, one containing data about projects, the other containing data about project administrators. You are welcome to do such a design if you want, but it is **not** mandatory for this exercise.)*

A9.2: Database Transactions

Destination file: **`A9/project_transactions.sql`**

Construct SQL queries for:

- **1)** showing a list of all projects in the database
- **2)** inserting a new project
- **3)** updating the project description for a given project.

Save your queries in the file **`A9/project_transactions.sql`**

A9.3: Site Map

Destination file: **`A9/site_map.gif/jpg/png/pdf`**

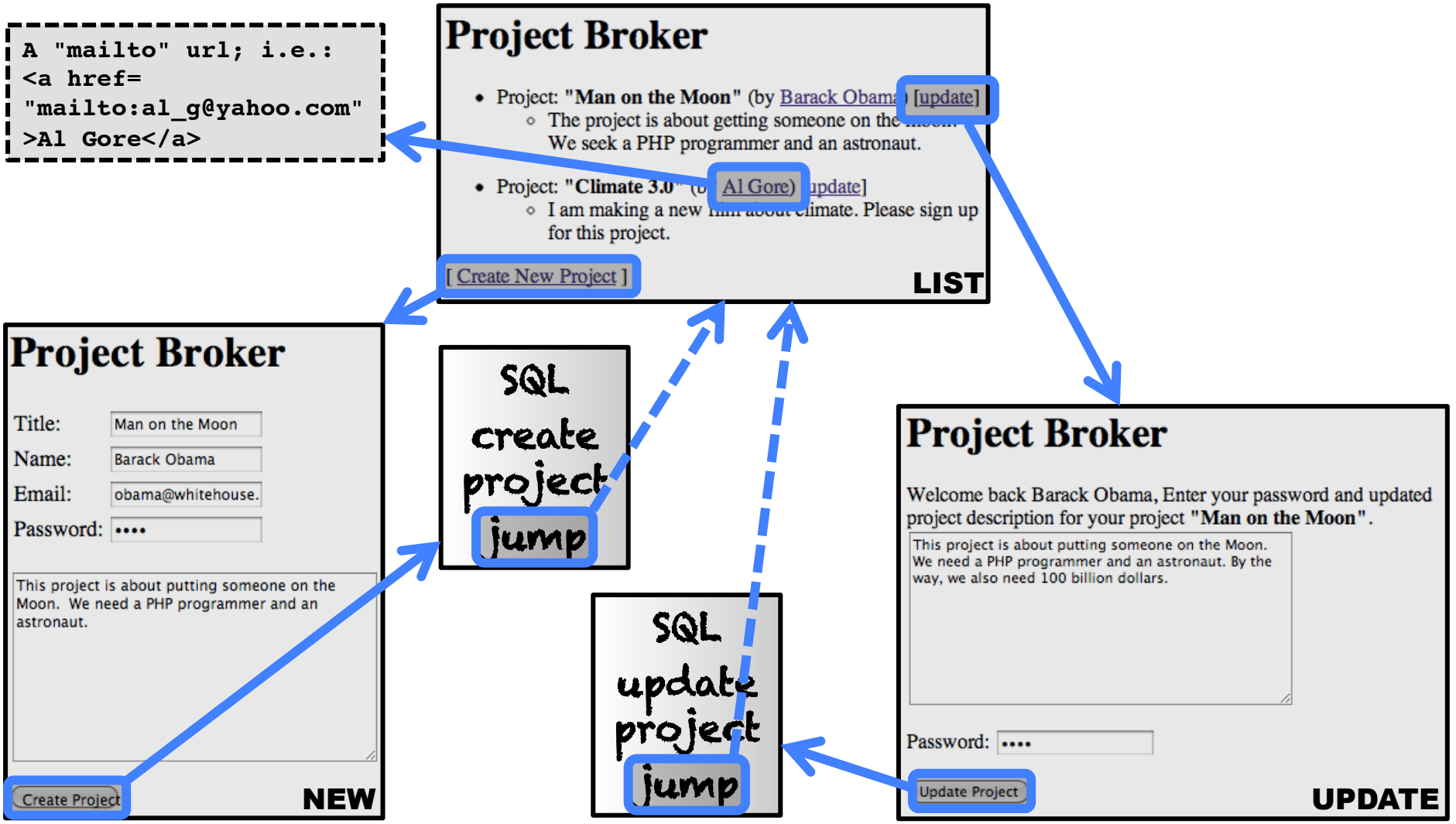
Now the time has come to design the overall sitemap of the broker service. What web pages and PHP scripts are needed and what variables need to be sent between them? On the next slide is a an almost complete site map diagram showing one possible structure of a Project Broker web service.

Your task is now to:

- **1)** to **give name** to all necessary PHP and HTML files (which are unnamed on the next page) in your project broker service solution; *and*
- **2)** to also define the **form variables** that **each file** is expecting to send and receive (if any). Please add that info next to the arrows.

The only constraint is that the main file (the one a user typically will access first) should be given the name **`A9/projects.php`** so that the TAs can quickly locate and start your service easily.

A9.3: Site Map (cont'd)



A9.4: Program Web Service

Destination files: **`A9/projects.php`**, **`A9/[yourfilenames.php`**, **`yourfilenames.html`**], **`A9/includes/`**[yourfilenames.php] (optional)

In this part of the exercise, you will implement (construct) the files needed to generate the web pages and performing the database transactions pictured in your design (A9.3).

Note 1: When implementing the PHP file for updating a project description, remember to include code for checking the project description password. Only if you know the password for the project, should a user be allowed to update a project description.

Note 2: You do not have to do any input validation in this exercise.

Note 3: The PHP scripts whose main task is to perform database transactions will probably benefit from using the PHP function **`header()`** to make the user's browser "jump" to a suitable web page after the transaction has been performed; e.g.

```
header("Location: projects.php");
```

Put all your PHP and HTML files for this exercise in **`A9/`**, and any files you include in **`A9/includes/`**.