Project

Minpeng Zhu
Form groups

- Groups of 4 (3-5)
- I want the following information from each group:
  - Names, ”personal numbers”, e-mail addresses
  - Contact person (”project leader”)
- Deadline for group formation:
  A week after course starts
Office hours / reachability

- Minpeng Zhu:
  - Email: minpeng.zhu@it.uu.se
  - Office: P1311
Project development events

- Write project plan
- Weekly meeting with the lecturer and internally
- Programming/coding
- Mid-term evaluation
- Hand in final report
- Present final presentation
Project plan

- Project plan deadline: Dates will be announced later.
- Project plan:
  - Motivation for your system
    - e.g. Why your system is needed? User group, age, etc
  - System function specification
    - ”What functions, services your web application provides?”
  - A system architecture
    - Must include
      - Use cases diagram + text description
      - ER diagram + text description
      - User interface description
  - An implementation plan
    - Functions priority list
    - Timeline of the project (what time finish what)
  - Group management
    - Task division, role assigned, etc.
  - Risks
    - Technical risk, time risk, etc.
Weekly meetings

- We will have a weekly 15 minute meeting for the project with each group, schedule 15 minute time slots with me
- You as a group are suggested to have at least weekly 15 minute long group meetings for the project

- Discussion:
  - Tracking the actual project progress with project plan
  - What is next step?
  - How to help each other in order to cooperate better as a group?
The date and time will be decided
  • Signup sheets for exact meeting times will be posted shortly
  • Exact time will be posted later

We will talk about the progress of your project
Presentations

- Dates will be announced later
- Presentation of your project
  - Usually, you decide what to talk
  - Presentation time is usually around 30 minutes per group (presenting + demo + answering questions)
- Demonstration of a working solution
Final report

- A written report, you might consider at least the following points:
  - Motivation for your system
  - System function specification (What functions your web application offers?, etc)
  - System architecture
    - Use case model + text description
    - ER model + text description
    - Screen shot from user interface + text description for some user cases
  - System implementation and evaluation
    - Development tools.
    - Technical problems encountered.
    - Experience gained.
  - Testing
  - Security
  - Group management
  - Conclusion and future work
  - Reference
  - Appendix
    - Figures
    - Source Code
    - etc

- Final report deadline
  - To be announced
How to do it?

- Think of a system you want to see
- Think about the use cases
  - How will the service be used?
- Cut the work into pieces
  - User Interface, application code, database
- Assign pieces within the project member
- Use Test Driven Development
  - Test each piece (unit tests)
  - Test all pieces together (integration tests)
Rapid prototyping

- Make a simple prototype ASAP
  - More new ideas might come up

- Enhance the system functionality step by step.

- Important to show the web application working flow according to your user case diagram.
Extra help, check out next slides
Recipes community use case diagram

1. Login with email and chosen password or with Facebook Connect function.
2. Enter email and choose a password to become member.
3. Enter title, text and image (optional). Tag with predefined tags, then the text is searched for terms matching.
4. Basic search looks first in title, then in description, ingredients and instructions, finally tags.
5. View a recipe.
6. Two sources, similar recipes and recipes you might like.
7. Shows the user’s personal information.
8. View a list of your recipes.
9. Rate a recipe on the scale 1-5.
10. Comment on recipe.
11. Allows the user to change his or her personal information.
12. Allows the user to remove a recipe he or she created.
## Use Case

<table>
<thead>
<tr>
<th>Name:</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Essential</td>
</tr>
<tr>
<td>Brief description</td>
<td>Users should register an account if they want to post comments or photos on the website.</td>
</tr>
<tr>
<td>Primary actors</td>
<td>Visitor</td>
</tr>
<tr>
<td>Precondition</td>
<td>None</td>
</tr>
</tbody>
</table>

**Basic path**

1. User clicks “register” link.
2. Enter register page.
3. Fill in the register form and click the submit button. The register form will include user email information, first name, last name, phone, password and user type.
4. Register process complete.

**Alternative path**

**Postcondition**

- Exception path starts at basic path 3 because of wrong email format.
  1. User inputs the wrong email format.
  2. System will give a prompt.

**Exception path**

- Exception path starts at basic path 3 because of unequal password.
  1. User inputs unequal password.
  2. System will give a prompt.

- Exception path starts at basic path 3 because of wrong phone format.
  1. User inputs wrong phone format.
  2. System will give a prompt.
Use cases examples:

- Register a new user
- Login
- Logout
- View profile
- Add, delete, edit protocol
- Make comments on recipe
- Rank recipe
- Search functionality
- ......
Webpage implementation technology:

- Asynchronous JavaScript and XML (AJAX)
- ASP.NET (build in validation and security control, paging formatting feature)
- PHP and HTML
  - Scripting language that is especially suited for web development and can be embedded into HTML
- .NET framework
- Cascading style sheets (CSS)
- Smarty Template (separate the HTML layout from PHP code, easy to change the layout of the website without the requirement to dig into the application code)
- Javascript (implement pop-up boxes, form validation, dynamic drop down list, etc)
PHP editor

- CodeIgniter (PHP web application framework)
- Bluefish (An editor with PHP syntax support, in-line PHP documentation, etc)
- Netbeans (A dedicated PHP coding environment and complete integration with web standard)
- ......
- Code sharing and version control:
  - CVS, SVN, dropbox, Github

- Document sharing:
  - Googledoc, ...
Data management

- MySQL DBMS
- SQL server
- J boss
- ......
Server

- WAMP (for Windows)
- LAMP (for Linux)
- Other server
ER diagram example from DB 1 course
Testing

- NUnit in Visual Studio
- JUnit in Java
- SimpleTest (PHP unit test and web test framework)
  
  - http://www.simpletest.org/

- ....
Simple PHP testing example

- Follow Model-view-controller (MVC) pattern
- Define a PHP class (model) for every table in the database
- An object of each model will represent a row in the corresponding table
- Example usage of the model classes:
  - $b = Book::getBookByISBN(1234512345);
  - echo “Book title is “ + $b->title
  - // Update the book
  - $b->title = “New Title”;
  - $b->status = 3;
  - $b->save();

- Create a test.php
- Example test (Add Book)
  - function testAddBook() {
  -   $book = Book::add(‘book title’, 0, 1234512345, ‘Good’, 200, 1, 2, 0);
  -   $book2 = Book::getBookById($book->getBookId());
  -   $this->assertEqual($book->getBookId(),$book2->getBookId());
  -   $this->assertEqual($book->isbn,$book2->isbn);
  -   }

  - }
Diagram design

- Tools:
  - ArgoUML
  - Microsoft visio
  - ...
Potential problems

- **ASP.NET**
  - University servers could not support webpages implemented by ASP.NET
  - Find another server to host your ASP.NET webpage.
General comments

- It will take longer time than expected to implement the project
  - Inexperience with these technology
  - Team work management