Team Projects

Internet Technology

ITECH

Overview of Team Projects

- **Design Specification (10%)**
  - Presentations need to be professional
  - Focus on communicating the design to the client
  - Show consideration for the users, include novel features
  - You will receive feedback on application design
  - Deadline: **06 Feb 2017**
- **Project Presentation (5%)**
  - Updated slides of your initial design specification
  - Your chance to demonstrate your app.
  - 5 minutes talk, 3 Q/A
  - Presentations will take place on **Week 10!**
- **Project application (25%)**
  - Deployed in pythonanywhere and source code in GitHub
  - Should be up and running in order to get your mark!
  - Deadline: **15 Mar 2017**

Know your Team!

- Your team should consist of 4 students (max)
- The team should be a mix of IT, CS, SD, …
- Your team mates will influence your final **INDIVIDUAL** mark…
  - So choose wisely!
- **Team Projects are like a 48 hours hackathon spread over 2 and a half months!**

Team Activity

- Your group will need to come up with an application to develop
  - You can find inspiration from the Design Briefs in ITECH's Moodle page!
  - Next week, we will go over a Design Specification from previous year

Team Activity

- Agree on a team name
- Application name
- What's the aim of your app?
  - Bullet points
  - Get a draft idea, you will refine it over the coming weeks

PROJECT BRIEFS

Overview of Briefs

- Rate This Course
- Participants Wanted
- Personalized Information Search Tool
- Requirements and Design Tracker
- Go Fish: a game
- Zombie Survival

RATE THIS COURSE

RTC Requirements

- The client would like to create a site to let student's rate courses that they take at the University.
- This app should allow users of the site to enter course details (if needed), and provide a rating and comments about the course.
- For course details it should include the course code, course name, course year and lecturer name.

RTC Specifications

- For the ratings, the user needs to enter the course, their overall rating of the course, and their comments.
- The site should let users find a course and see all the comments and ratings
- It should let users see a list of the latest ratings/comments
- It should let users see a list of the best/worst rated courses
- The front page should present the latest comments, the best/worst rated courses, and some form of search/navigation of courses

RTC Specifications

- To register students will need to provide a valid university email
  - The email should be confirmed, before they can add ratings.
  - The admin of the site would be like to be able to add in different universities, and the email domain for legit universities.
- Courses can be added to the system by students
  - Some check to make sure the course has not already be added should be included

PARTICIPANTS WANTED

Participants Wanted

- Need to run an experiment but need willing participants?
- Want to earn some extra cash, or would like to learn more about research?
- Tries to match participants and experiments.

PW Requirements

- Students at a university should be able to register with the PW system, and complete a demographics survey (age, sex, first language, home country, education level, email, etc)
- Students will be able to browse through the list of current experiments (which meet any requirements i.e. age, language etc),
- When they identify an experiment that they would like to participant in they can select it, and see the details of that experiment.

PW Requirements

- The student can then offer to undertake the experiment, by expressing their interest (like making a bid)
- The student will be able to look at available experiments, past experiments that they have undertaken, and experiments currently bid on.
- If a student's bid is accept the student is notified (through the interface and via email)

PW Requirements

- It would be nice if the system could
  - send out the list of up and coming available experiments
  - Offer slots/times when they could undertake the experiment
- An experimenter should be able to register with the system (entering in their details, name, title, school, web page, contact info, etc)

PW Experimenter Reqs

- The experimenter should be able to enter in an experiment
  - (start and end date,
  - accepting bids (true or false),
  - number of participants required,
  - number of participants so far,
  - experiment title, experiment details, and possibly a link to more information)
- The experimenter should be able to view bids, screen participants anonymously, and then accept or decline the participants offer.

PW Main Page

- The main page should encourage the participants to register
  - Provide a login box
  - Link to experimenter page/view
  - If signed in, link their profile page
  - If signed in, they should get a notification of any experiment that they have been accepted for
- It should show summaries of the last 5 experiments entered,
- It should show 5 summaries of experiments with up and coming closing dates,
- And 5 summaries of experiments which pay the most reward (and that are currently open).

PW Other requirements

- It should be possible to view all available experiments (ordered by latest, closest and payment) – user do not need to be logged into to view, but then do if they want to bid.

Variation

- Instead of trying to find participants, the same application could be called InternFinder
  - Where companies are looking for interns
  - And students are looking for internships

PERSONALIZED INFORMATION SEARCH TOOL

Search Tool Requirements

- Choose a search engine (Bing, Flckr, Arixv.org) to create a search tool for web, news, photos, papers, etc.
- Users of your search service should be able to register/login – so that their searches can be saved.
- Users will be able to create a topic – which they will then conducted searches, and save documents to.
- All queries issued when searching for information about that topic should be recorded – so that they can go back and revisit previous search results.

Search Tool Requirements

- User should be able to mark documents as interesting/useful
- For each topic they should be able to see
  - what queries they performed (and be able to delete queries that were not useful)
  - and what documents that have marked useful
- They should be able to see the list of topics that have saved, they can add/edit/delete topics

Search Tool Requirements

- Would like features include:
  - The system should highlight results previously seen
  - The system should exclude showing documents they have marked not useful
  - The system should be able to exclude results from certain domains
  - The system should be able to only show results from certain domains
  - The system should let users share documents with others via social media (i.e. tweet this link).

REQUIREMENTS TRACKER

Requirements Capture Tracker

- Creating the specifications list and capturing the requirements of a system is collaborative and ongoing process
- The aim of this application is to help collect the these requirements and specifications.
- This is like a collaborative TODO list

RCT Requirements

- A developer should be able to register, and create a project
- Other developers can be added to the project
- A developer's page shows an overview of the list of projects that they are working on
- On a project page a developer can see the list of requirements already taken, and add other requirements

RCT Requirements

- The project page should look like a board, where the specifications can be grouped into different categories (must/should, could would like), and ordered by priority.
- When a specification is entered, a date/time is associated with it, it has a priority level (1 to 5, 1 being the highest), and has a category (as described above), and a short description.
- The specification is associated to a project

RCT Requirements

- Would like features to include
  - A way to track or mark versions so that it can be possible to view previous versions and how the projects have been iterated.
  - Specifications can be dragged between categories and ordered

GO FISH

Go Fish Game Idea

- The game is based around the idea of fishing
- A player is assigned a certain amount of time to go fishing
- They select a spot to fish (this takes time), and then decide how long to fish for (which also takes time)
  - For example it might take 30 minutes to go to a fishing site
  - Then they can spend 15 minutes fishing
  - Then can then decide whether to keep fishing, or move to another site
  - After they fish, they are informed as to how many fish they have caught

Go Fish Requirements

- The game continues until the player is out of time
  - The number of fish they catch, determines how much money they get at the end of the game
  - Money is used to buy upgrades
    - Upgrades consistent of better boats (which makes travel time cheaper)
    - Scanners to indicate if fish are in the patch

Go Fish Requirements

- A game (round of fishing) consists of a number of fishing patches
  - Each patch has a payoff (number of fish)
    - Depending on how long the player spends depends on how many fish are caught for instance
      - Payoffs = [0, 0, 1, 3, 0, 0, 2, 0, 0]
      - First fishing 0 fish, second 0, third 1, fourth 3, and so on.
    - These patches can be varied to change the difficulty and challenges

Go Fish Requirements

- The game can be text based, graphical, or some kind of representation
- The game should keep track of the user's progress through a series of games.