

D4 Assignment Briefing: Implementation & Demo

This “D4” assignment relies on the case study briefing documents provided with “D2”, “D3” and any subsequent amendments, which outlined the background to the proposed prototype “Staff Access Management System” (SAMS-P). Where there are differences the most recent document applies, i.e. this document has precedence over all the others. To get started, you should:

1. read the introduction below,
2. read the “e-mail” in Part 1 of this document,
3. then read the Design Study in Part 2,
4. then continue with the tasks specified in Part 3.

Introduction: Goals

This assignment is designed to give you practice in software engineering techniques. In particular, you will experience some of the technologies and challenges often found in contemporary real-world software development.

The **technologies** include development platforms, frameworks, web-based architectural design (including MVC), object-oriented design, php, sql, plus html, css, and javascript.

The **challenges** include:

- teamwork, including leadership & negotiation (building on your experience in D2 & D3)
- planning, time management, and fixed deadlines
- demonstrating software to a notional “business client”
- coping with a complex (possibly incomplete and inconsistent) brief
- interacting with a client
- reading technical documentation and obtaining technical support from a “software supplier”

Introduction: Tasks

- refine an existing requirements specification and design,
- further develop the design, e.g. with a data model and GUI (graphical user interface)
- document the data model
- develop software using the supplied framework
- test the software
- demonstrate the software to the client
- carry out project planning & management
- deliver an analysis of the management of the project as a slide presentation.

Introduction: Assessment

This is primarily a **group-work** exercise, worth 15% of the total for the module. For details of the assessment see Part 4 (section 4.1).

With group-work, all team members are expected to make equal contributions. These can take different forms: not just volume of work, but also specialist skills, knowledge, or leadership. You should plan for approximately equal contributions and we expect most teams will achieve this goal.

However it does not always work out like this, and so in D4 there will also be an **individual assessment**: to check that everyone has contributed, and to what extent. This is worth 5% of the total marks for the module and will take the form of peer-group assessment moderated by the module team (the demo examiners). For details, see Part 4 (section 4.2).

Part 1: e-mail Briefing *[the scenario]*

From: **Ms Penny Farthing, Business Development Director ITMan Systems**

Date: **05-Jan-2012**

To: **your Team**

Subject: **Phase 3: SAMS Prototype Project**

As you know ITMan has won a contract to produce one of the prototypes for the “Staff Access Management System” for the London 2012 Olympics. I would like to thank you for the work that you have done so far in producing the Requirements *[D2]* and Design Specifications *[D3]* for the SAMS Prototype.

As a result of your work I am pleased to inform you that ITMan has been short-listed, and invited to a final run-off. We have been asked to develop part of the SAMS Prototype and demonstrate it to the client.

I would like your team to take responsibility for the development and the demo.

Part of our bid is to showcase the capability of our newly adopted standard development platform, MoWeS, and the VF1 PHP Development Framework that runs on it. Hence you must base your work on them. I appreciate that they may be new to you but documentation has been provided and our platform/framework supplier, WhatTheHeck Ltd, has promised to provide full technical support. The advantage of this approach is that you will only be required to edit some code, not create complete programs from scratch. The underlying technologies are industry standards: HTML, CSS, PHP, JAVASCRIPT and MySQL.

For the demo, you will be provided with what is effectively a **User Acceptance Test (UAT) script**. It goes without saying that the demo must run faultlessly and so you should test your software thoroughly in advance. The UAT script will rely on a mix of pre-defined and ad hoc test data, the latter determined on the day.

The demo will consist of the following steps designed to convince the client *[examiners from the module team]* of the quality of our development processes and management:

- live running your software against the UAT script, plus
- a short presentation, plus
- a Q&A session.

It has now been agreed with the client that the software for the demo will focus on the capability of the SAMS prototype to support the administration of authorising staff access to Olympic venues. In other words the focus will be on how the proposed application will be used by administration and reception staff to support venue authorisation and to log/archive staff access. For the purposes of the demo, the only functionality to be implemented will be for the management of staff authorisation and access to Olympic venues. The following information needs handling:

- third party staff, e.g. from employment agencies (when applicable),
- in house Olympic staff,
- authorisation cards,
- venue details,
- staff access per venue (a facility is needed to log these).

For the purposes of the demo, **the users** to be considered are the admin staff at the Staff Central ID Office, or SCIO.

This implements part of the design you should have already produced *[D3]*; you are free to incorporate your existing design (allowing for any changes) or not, as you wish. One additional task you will need to do is design the data model, which must be compatible with the framework supplied.

Extensive documentation of the task, written by me, is attached to this email [Part 2 of this briefing]. I am very busy at present with other aspects of our bid, but I am prepared to respond to any further questions you might have **about the client's brief**.

In addition, **Technical Support** for the MoWeS platform and the VF1 PHP Development Framework is available, but from a different source: the supplier, WhatTheHeck Ltd.

Part 2: Design Study *[background]*

2.1 System Outline

It has been decided that staff access authorization will be based on the assumption that a card-reader system *[outlined in D2]* will be in place by the entrance of the venue. However, for the purpose of the demo, it should be possible to enter manually the card id number within the system. The system will then check whether and when (from... to...) that card is authorised to access that venue (this should be reflected by having set the card status to “valid”). The system should log whether an access has actually occurred (this should be reflected by having a new access record in an access log table).

Other possible states of cards can be “expired” and “cancelled”: the “expired” status should ideally be automatically set by the system at the end of the validity period for that card. The “cancelled” status means that a card validity status has to be manually removed by a member of the admin staff, for instance, if that card is lost.

An ideal constraint of the system should be that no staff should be given more than one valid card at once.

2.2 Data Model and Description

The database core of the SAMS-P prototype for administration of staff access should be made up of five “main tables” (i.e. those tables that map the classes of objects identified at application level: “agencies” (or “employers”), “staff”, “card”, “access” and “venue”), and at least one join table to support the many-to-many relationship between cards and venues underlying the authorising process.

Other tables, such as those holding supporting complementary information, are usually expected from the VF1 framework, although they do not need to be directly managed by the application interface (for instance, the “status” table, i.e. a table containing all the possible states of a card, as specified above).

All these tables should accommodate data about agencies, staff, their cards, the venues, and – obviously – access authorisations. The tables should also accommodate accesses that have occurred (and been logged). You will need to design and implement the names, structures and some specimen data for the tables, taking into account the requirements outlined in this document, as well as the constraints imposed by the VF1 framework. The column names should be chosen to be self-explanatory.

It is important that you design the relationships between entities of the database in such a way that your software will implement, on top of all the CRUD operations on records of the main tables, the following scenarios:

- a) authorising a new card for a new member of staff enabling him or her to access an existing venue
- b) retrieving all authorised venues for a member of staff in a given period of time
- c) retrieving all logged accesses for a member of staff to a given venue

There should be an efficient workflow for data entry.

Please, be aware that the data model expected here is not necessarily the ideal one (actually, it is not), but it has to be like this because of some implementation constraints related to the MyActiveRecord class used in PHP and the VF1 development requirements.

Specimen data will be provided in due course and should be loaded into the database for the demo.

2.3 The MoWeS Platform and the VF1 PHP Development Framework

The MoWeS platform may be downloaded from the following location:

http://homepages.stca.herts.ac.uk/~comqv/wos_csitde.zip.

The VF1 framework, although already included within the MoWeS platform above, can be independently downloaded and further information obtained from the following location:

<http://homepages.stca.herts.ac.uk/~comqv/vf1>.

The home pages for MoWeS and VF1 with access to the software and the documentation may also be found at <http://homepages.stca.herts.ac.uk/~comqv/csitde/d4.html>.

See also the supporting teaching material posted on Studynet as lectures or tutorials.

A MySQL database is provided within the MoWeS platform. Because of some design constraints related to the “MyActiveRecord” class and the VF1 development framework, the following conventions apply:

1. Table/Class mapping is achieved by each database table being named **identically** to the MyActiveRecord subclass that will represent it (in lowercase for compatibility reasons): for example, the class “staff” will be mapped on the table “staff”.
2. Primary keys in main tables are always auto-incrementing primary keys and named “id”
3. Foreign keys as might be identified in the tables are named as follows: “tablename_id”. For example, a foreign key to the class “airline” in the table “flight” will be named “airline_id”. When two or more foreign keys to the same table are needed (meaning two or more different relationships from one table to another one), foreign_keys should be named as follows: “relationship1_tablename_id”, “relationship2_tablename_id”, etc. For instance, “flights” might have two foreign keys to the destination “location” table, namely “from_location_id” and “to_location_id”.

Part 3: Assignment Tasks and Expected Outcomes

3.1 Deliverables

3.1.1 For your coursework submission

You should produce and submit the following three deliverables, according to the process and submission date given in the Assignment Briefing.

- your software (soft copy on CD)
- your design documentation (printed copy)
- your presentation (printed copy plus soft copy on CD)

3.1.2 For your demo (at a later date)

You should bring to the demo (the process and schedule will be published in due course)

- your software, able to run with minimal set-up, e.g. on USB flash memory
- your presentation, able to be displayed on a PC with minimal set-up
- your completed “D4 Roles and Contributions Form”, (individual forms if you disagree as a team; see later); *there is no need to include this with your coursework submission.*

At the demo, your software and presentation must be identical to that submitted on the CD.

3.2 Test Data

Specimen test data will be provided well before the demo. These data are expected to be details of six staff, their card(s), venue(s) and access authorisation, and access occurrences.

Your application database should be able to store all the provided test data, or equivalent, before your team's scheduled demo. This is because this test data set will be used to assess your software against the “User Acceptance Test” checklist shown in Part 4 of this document.

While you are testing your system for validation/verification purposes prior to the demo you may of course use different data. However please be advised the version of the application submitted for demo purposes must have data entered that conforms to the specimen test data.

3.3 The Software

Your team will be expected to demonstrate that your software meets the functional and non-functional requirements specified in this briefing. Some examples are shown in the following sections.

3.3.1 Functional Requirements

The system should demonstrably support:

- The input of additional employment agencies, staff, cards, authorisations, accesses and venues
- The amendment of agencies, staff, cards, authorisations and venues
- The retrieval of information about agencies, staff, cards, authorisations, accesses and venues

It is expected that, in general, the system functionality will support managing staff venue access in as flexible a way as possible. In other words searching, selecting and amending individual records for agencies, staff, cards, authorisations, accesses and venues will be possible in different ways.

As an example of flexibility in retrieval, searching for an individual staff authorisation could be done on either staff name or card number.

As an example of flexibility in amendment, data on authorised access dates can be changed, but data logging occurrences of venue accesses cannot.

In the above examples, user tasks could be more efficient if it were possible to combine these single actions with other actions such as changing authorisation for access to existing venues, or by granting access to further venues. See also Section 2.2, “Data Model and Description”.

Please note that, during the demo, the UAT checklist (see section 3.2 Test Data) will be used by the examiners to establish specific input/amendment/retrieval details.

3.3.2 Non-Functional (Usability) Requirements

Input or amendment of access authorizations etc should be coherent and contain a linear sequence of actions: an efficient and effective workflow for the target users.

For example, a form to enter an agency's details (to be stored in the respective table), should confirm the data entry made. Once a member of staff exists on the system, it should be immediately possible to authorize his or her access to a venue by appropriate navigation. For example, it should be possible to go easily to another form to input staff details (to be recorded in the appropriate table), thereby granting access to any authorised venue (again, to be recorded in the appropriate table).

Other expectations for the user interface are that it satisfies established HCI (usability) good practice. For example, once a new member of staff has been entered into the database, the system should give immediate feedback of that data entry, for example by providing some basic information (such as staff id, full name, etc.).

Other HCI good practice expected includes the ability of the interface to be both easy and efficient to use (by its target users). For example the user interface should provide alternative views of the database, from displaying an existing single record to being able to display other records already entered into the database. Such views of the data should be easy to display and update with minimum effort by the end-user, for example in "one click".

3.4 Assessment of the Software

The SAMS Prototype should consist of a set of fully working .php files plus any other .css and .js files specifically developed or customized.

The SAMS-P application should be connected to a "real" database (in the MoWeS platform, MySQL) that should be accessible independently from a database management system (DBMS) (in the MoWeS platform, phpMyAdmin). This will permit the examiners to verify during the demo that actions performed by the application have the appropriate effect on the database.

Specimen test data for the D4 assignment to be used within the MoWeS platform will be provided on StudyNet.

The assessment of your SAMS Prototype will take place at your team's demo; marks will be awarded according to how well your system is able to satisfy the functional and usability (HCI)-related requirements given above.

See the marking scheme in Part 4 for further details.

3.5 Design Documentation

You only need produce documentation for your database design; this should consist of:

- an E-R diagram
- the data dictionary

The diagram should be drawn using suitable specialist software or a standard drawing tool. Hand-drawn diagrams will be accepted providing they are legible.

3.6 Support

Support is available from the ITMan Business Development Director to clarify the **client's business requirements** and from WhatTheHeck Ltd support team for **technical information** as follows.

*To contact "Ms Farthing" to answer questions about **THE CLIENT'S BRIEF** you should use the **Class Discussion Board** on the Studynet site for the module.*

***You must post to a thread prefixed "D4: ask Ms F - ..."** for example "D4: ask Ms F - target screen size". You can start a new thread if you like but be sure to check all existing threads first. Ms Farthing does not like answering questions she has already answered.*

*To contact the software supplier to obtain **TECHNICAL SUPPORT** for MoWeS or VF1 you should use the **Class Discussion Board** on the Studynet site for the module.*

***You must post to a thread prefixed "D4: ask WhatTheHeck - ..."**. You can start a new thread if you like but first be sure to check all existing threads and the documentation already posted on the website. WhatTheHeck are entitled to charge for information they have already provided.*

In fact, these are the only ways to get support. We will not accept requests by email for new information nor will guidance be given in the labs or in person - since, to be fair to everybody, we wish to make such information visible to all. We may also post updated information from time to time.

3.7 The Presentation

As part of the demo, you should deliver a short (under 5 minute) presentation with 5 slides maximum to explain the processes you used to ensure the quality of your software and the management of your project. You should cover:

1. your team working;
2. your project management;
3. your test plan and outcomes;
4. evaluation of your system's usability;
5. overall analysis and reflection: lessons learnt; what we did right and what wrong

You should produce the slides in advance (in a standard format such as Powerpoint) and submit them with the software (see the Assignment Briefing).

Note that the presentation should relate to your actual experience; this is **not** a role play.

3.8 The Demonstration

See the Assignment Briefing Sheet. Administrative and other details will follow nearer the time. Watch the Studynet News for details.

It is your team's responsibility to ensure that your SAMS prototype (including the application and the database) and an independent DBMS (i.e. phpmyadmin, which is provided within the MoWeS platform) will be ready for use and fully working during the demo.

The demo will be hosted on any one of the PCs in the CS labs and so you should test your software on this platform in advance. You are advised to use Firefox as your web browser since it is standards-compliant.

Personal laptops are not to be used for the demonstration, unless explicitly authorised by the module leader in advance. You are reminded to check regularly the module pages on StudyNet for further details about the demo timetable and other arrangements.

Part 4: Assessment

4.1. Marking Scheme – Teamwork

The teamwork is worth 15% of the mark for the module, with each team member awarded the same score. During the demonstration, your team will be assessed against the following criteria:

4.1.1 The Outcomes of the User Acceptance Test (UAT)

this assesses your team's design, coding and testing skills and forms a checklist that will be used on your team's marking & feedback sheet. Please also refer to section 3.2 test data.

1) The SAMS-P system allows the display of details from six staff members:

- where entries are stored already in the database yes (4) no (0)
- where entries are viewable and editable by the system interface yes (4) no (0)

Item Total: out of 8 marks

2) The SAMS-P system allows the insertion into the database of a new authorisation and the registration of a new agency and a new staff member:

- by adding details into the "staff" table yes (4) no (0)
- by registering a new card for that staff member yes (4) no (0)
- by setting up the dates of validity for that card yes (4) no (0)
- by displaying and authorising access to available venues yes (4) no (0)
- by setting the status to "valid" yes (4) no (0)

Item Total: out of 20 marks

3) The SAMS-P system allows the updating of an existing access so that its status is changed to "cancelled":

- by searching for authorised access to venues by staff members yes (4) no (0)
- by visualizing which venues that staff member's card is authorised to access yes (4) no (0)
- by possibly resetting the dates of validity for that card yes (4) no (0)
- by displaying all available venues yes (4) no (0)
- by authorizing that staff member to access a different subset of those venues yes (4) no (0)

Item Total: out of 20 marks

4) The SAMS-P system allows data to be retrieved and displayed containing the following information:

- all the staff members who can access a given venue on a certain date yes (4) no (0)
- all the venues accessible by a given staff member yes (4) no (0)
- all the venues actually accessed by a given staff member on a certain date (chosen by the examiners during the demo) yes (4) no (0)

Item Total: out of 12 marks

5) The SAMS-P system supports the following:

- it prevents the issuing of a new authorised access to a venue if it has already granted access to that venue on that date yes (4) no (0)
- it prevents registering the same staff member more than once yes (4) no (0)
- any extra useful functionality, based on the brief but defined by you: [up to 8]

Item Total: out of 16 marks

SUB-TOTAL (UAT): out of 76.

4.1.2 The User Interface

this assesses the usability of your system

6) The SAMS-P user interface will be assessed against the following usability criteria:

- system navigation is clear, easy to move between edit boxes, efficient workflow [up to 8]
- quality of the overall look and feel (includes: naturalness; predictability; layout and colours consistency; sensible design: use of colour, fonts and size, controls easily identifiable) [up to 4]
- feedback is provided after every user's action (acceptance of entered data is always confirmed and system status is given) [up to 4]
- error messages are meaningful and helpful [up to 4]
- basic on-line help is available [up to 4]

SUB-TOTAL (UI): out of 24.

4.1.3 The Content of the Presentation

this assesses your analysis and reflection of the management of your project

7) The presentation will be assessed against the quality of the following topics; your analysis and reflection on:

- team working: *includes a reflection on how you worked as a team* [up to 6]
- project management: *includes a reflection on how you managed the project* [up to 6]
- test plan and outcomes: *includes the design of your test plan and the outcome of the tests* [up to 6]
- evaluation of your system's usability: *includes how you ensured that your system met the usability criteria in practice* [up to 6]
- overall analysis and reflection (lessons learnt): *includes an honest analysis of what you did right, what you did wrong, and how you could improve "next time"* [up to 6]

SUB-TOTAL: out of 30.

4.1.4 Quality of Delivery of the Presentation

this assesses your communication skills

8) The demo and presentation will be assessed against the quality of the following:

- quality of delivery of the demo: *includes preparedness, timekeeping and professionalism* [up to 5]
- quality of delivery of the presentation: *includes presentation skills, timekeeping, and design of the slides* [up to 5]

SUB-TOTAL: out of 10.

4.1.5 Quality of Documentation

this assesses your design and documentation skills

9) Your documentation will be assessed against the quality of the following:

- E-R diagram and data dictionary [up to 10]

SUB-TOTAL: out of 10.

OVERALL-TOTAL: out of 150.

The total mark will be scaled to count as 15% of the mark for the module. Marks will be given as a percentage and then scaled to the nearest half integer out of 15 i.e. 0.0, 0.5, 1.0, 1.5, ..., 15.0.

4.2. Marking Scheme – Individual

The individual contribution is worth 5% of the total for the module, i.e. a maximum of 5 marks. It is “peer-assessment” moderated and adjusted, if necessary, by the examiners.

During the demonstration, you will also be examined individually to assess your personal contribution to D4. This takes the form of “peer assessment” prepared in advance by you and your team and individual questions or requests made by the examiners at the demo.

4.2.1 Peer Assessment - Preparation

As a team you should consider the “D4 Roles and Contributions Form” posted with the assignment. This asks you to describe the role of each individual and provide a percentage “contribution” that each has made to D4 (including preparation for the demo). For example:

Example (a): if there are five team members and you consider that everybody has contributed equally, then everyone should be allocated 20.0%. The total must of course be 100%.

Example (b): if there are four members and you all agree that one person has contributed twice as much as everybody else, then he or she gets 40% and the other three get 20% each.

If you all agree then you should complete a single team form and bring it to the demo.

If you cannot agree, then each team member should complete the form individually instead.

4.2.2 Questions at the Demo

The examiners will ask each individual team member one or more questions during or after the demo – and may ask specific individuals to demonstrate part of the software. Failure to respond satisfactorily may affect your individual score since it will indicate that you did not fully engage with the team.

4.2.3 Scoring

Each team is allocated a “pot” of 5 marks * <the number of team members> * <the percentage score for the teamwork>, i.e. *the better your team score the more individual marks available*. So, with the above examples, in §4.2.1:

Example (a): if the teamwork score is 50%, the pot is $5 * 5 * 50\% = 12.5$

Example (b): if the teamwork score is 80%, the pot is $5 * 4 * 80\% = 16$

4.2.4 Scoring - normal case

If all team members agree on roles and contributions – and answer all the questions satisfactorily, then the pot is simply divided up depending on the percentage contributions that you agreed. So, *if you have contributed more, you get more marks and, if you have not fully contributed, you will get fewer*. With the above examples, in §4.2.1:

Example (a): Everybody gets $12.5 * 20.0\% = 2.5$ marks (which is 50% of the 5 marks available)

Example (b): The first person would get $16 * 40\% = 6.4$ marks, the other three $16 * 20\% = 3.2$ each

However 5 is the maximum and so the first person is capped at 5, the other 1.4 marks are divided up amongst the other three, so:

Example (b): The first person actually gets 5 marks, the other three $3.2 + 0.47 = 3.67$ marks each.

4.2.5 Scoring – exceptions

1] If all team members agree on roles and contributions – but not all the individuals answer their questions satisfactorily, then the examiners reserve the right to adjust the scores for those individuals, *including the possibility of awarding zero marks*, depending on:

- individuals’ roles as outlined on the form;
- quality of responses to the questions.

2] If all team members do not agree on roles and contributions, then the examiners will arbitrate, depending on:

- any majority consensus;
- roles as outlined on each individual’s form;
- quality of responses to the questions.

Part 5: Assignment Checklist

This is a summary of the main tasks for you to use as a checklist; full details are given elsewhere.

Task	Item	Done
1	Read all the documentation thoroughly.	<input type="checkbox"/>
2	Make a project plan . As a minimum, this should: <ul style="list-style-type: none"> - incorporate all the specified tasks; - include some intermediate milestones so you can assess progress before it is too late to recover; - allow for some contingency in case anything goes wrong (technical or people issues); - allocate tasks to individuals; decide how you will integrate the contributions. 	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
3	Carry out all the tasks , including: <ul style="list-style-type: none"> - develop your software - produce a test plan - test your software against the test plan (including the "UAT"), using the specimen and other test data - review, make any corrections, re-test, and repeat. <i>Note that you are likely to find this stage much easier and to get a better result if you attend the lectures and lab sessions designed specifically to support these activities.</i>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
4	Produce the deliverables for the coursework hand-in : <ul style="list-style-type: none"> - your software, including the specimen test data (on a CD); - your design documentation (printed copy); - your presentation (printed copy and on the CD). 	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
5	Sign the Briefing Sheet. All individual group members who contributed must do so; no signature means no mark.	<input type="checkbox"/>
6	Check your completed solution meets the Assignment Briefing . Package your solution with the completed Briefing Sheet in a transparent plastic folder with your team id clearly visible and the CD securely stored within.	<input type="checkbox"/>
7	Submit your coursework using the STCA Automated Assignment Facility by the specified date and time shown on Studynet.. Affix the submission sticker to the front-sheet, NOT to the plastic wallet (assuming the label printer is working).	<input type="checkbox"/>
8	Submit ONE zipped file containing ALL of your key files for the demo, including your software, database, and demo slides (but not the MoWeS platform). This should be by the same deadline as the hard copy/CD hand-in. <ul style="list-style-type: none"> - only ONE submission per team is needed, to a group area on Studynet; - it will only be used for backup or quality assurance purposes; - responsibility for this submission is with the team as a whole. 	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
9	Prepare for the demo : <ul style="list-style-type: none"> - establish the time and the place for your demo and ensure all team members attend; non-attendance means no marks for the individual; - test that your software works on the lab PCs with Windows 7 and Firefox; - allocate roles and practise both the software demo and the presentation. 	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
10	Produce the deliverables for the demo : <ul style="list-style-type: none"> - your software, ready to run (e.g. on USB flash memory); - your presentation slides, ready to display; - your completed D4 Roles and Contributions Form(s). 	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
11	Attend the demo . Then it's all over! Wait for the written feedback, due after all the demos have finished.	<input type="checkbox"/>